

**KeTop T50**  
**Handheld Terminal**  
**User's Manual V 1.6a**



Automation by innovation.

## Notes on This Manual

At various points in this manual you will see notes and precautionary warnings regarding possible hazards. The meaning of the symbols used is explained below.

### **DANGER**

- **DANGER** indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

### **WARNING**

- **WARNING** indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

### **CAUTION**

- **CAUTION** indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

### **CAUTION**

- **CAUTION** used without the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property injury.



This symbol reminds you of the possible consequences of touching electrostatically sensitive components.

#### **Note**

*Notes on use of equipment and useful practical tips are identified by the "Notice" symbol. Notices do not contain any information that draws attention to potentially dangerous or harmful functions.*

© KEBA 2006

Specifications are subject to change due to further technical developments. Details presented may be subject to correction. All rights reserved.

Document: version 1.6a / material no.: 61139  
Filename: t50\_en.doc, last saving on: 25. 10. 2006

KEBA AG, Postfach 111, Gewerbepark Urfahr, A-4041 Linz  
Tel.: ++43 / 732 / 70 90-0, Fax: ++43 / 732 / 73 09 10, E-Mail: keba@keba.com, www.keba.com

KEBA GmbH, Ulmer Straße 123, D-73037 Göppingen  
Tel.: ++49 / 7161 / 97 41-0\*, Fax: ++49 / 7161 / 97 41-40

KEBA Corp., 100 West Big Beaver Road, Troy, MI 48084  
Tel. ++1 / 248 / 526 - 0561, Fax: ++1 / 248 / 526 - 0562, E-Mail: schr@us.keba.com

## History

Modification from / to	Date	Modified pages	Description	Author
V1.0	28.6.02		first edition	sam
V1.0/V1.1	Juli 02	28ff	Chap. „Shielding“ has been added.	sam
V1.1/V1.1a	17.7.02	23	Chap. „Example of Connection with PILZ PST1 Control Relay“: Instead of 2 diodes and 2 resistors a symbol for motor has been added.	sam
V1.1a/V1.2	2.8.02	11	Chap. „Elektronic“: RS-232-C interface has been added.	sam
		13	Chap. „Power supply“: The notice „Maximum interruption time of supply voltage:“ has been added.	sam
		31	Chap. „Cable entrance area“: Drawing has been changed..	sam
		33	Chap. „Cable outlet“: new drawing	sam
		34	Chap. „RS-422-A“: has been changed (Dip switches, external device, ..)	sam
		36	Chap. „RS-232-C“ has been added.	sam
		38	Chap. „Labelling of keys“: The drawing „Inserting of a paper label,“ has been added.	sam
		39	Chap. „Labelling of keys“: The drawing „Dimensions of labels in mm (scale 1:1)“ has been changed.	sam
		42	Chap. „Selection of Protocol“: has been extended.	sam
		48	Chap. „Display Character Set“ has been changed..	sam
		52	Chap. „Diagnosemenü“ erweitert (Display und COM Test).	sam
		55	Chap. „Setup Menu“ has been extended (Parameter)	sam
		55	Chap. „Info Menu“ has been extended (Software)	sam
		56	Chap. „System Errors“ has been extended (Time)	sam
		60	Chap. „Connection diagram“: has been changed.	sam
		65	Chap. 11 „Transport conditions“ has been added.	sam
		66	Chap. 12 „Technical data“: „Vibration resistance“ has been changed and „Maximum interruption time of supply voltage:“ has been added. Interfaces has been extended by the RS-232-C.	sam
V1.2/V1.3	1.10.02	-	All notes and precautionary warnings regarding possible hazards has been changed and adapted to ANSI z535.4. The new notes are described on page 2.	sam
		59,61,62	The following colours of the connection cable and the intermediate cable has been changed: S19, Pin 6: red => pink, S19, Pin 8: green-brown => brown-green, S19, Pin 9: weiß-grau => weiß-grün	sam
		65	Chap. "Technical data ": The storage temperature has been changed: OLD: -25 °C to +70 °C NEW: -20 °C to +70 °C	sam
V1.3/V1.4			New notation: KETOP -> KeTop	sam
V1.4 / V1.5	Juni-2003	alle	small changes	sam
V1.5/V1.6	Dez-2004	alle	new layout	sam
V1.6/V1.6a	10-2006	130	Chap. "Connection Cable KeTop TTxxx": The drawing has been corrected regarding the cable length	sam



## Contents

<b>1</b>	<b>Brief Description .....</b>	<b>7</b>
	Construction.....	8
	Ergonomic Features .....	9
	Housing.....	9
	Operating and Display Panel .....	9
	Hardware .....	10
<b>2</b>	<b>Safety Precautions .....</b>	<b>11</b>
	Power Supply.....	12
	Emergency Stop Switch.....	13
	Enabling Switches .....	13
<b>3</b>	<b>General Instructions.....</b>	<b>16</b>
	Intended Use of KeTop .....	16
	Waste disposal .....	16
	Handling of KeTop .....	16
	Foreseeable Misuse of Enabling Switch.....	18
<b>4</b>	<b>Safety Elements .....</b>	<b>19</b>
	Enabling Switch .....	19
	Emergency Stop Switch.....	25
<b>5</b>	<b>Shielding.....</b>	<b>26</b>
	Shield Connections.....	26
	Shielding inside the Control Cabinet.....	28
<b>6</b>	<b>Connection .....</b>	<b>29</b>
	Cable Entrance Area .....	29
	Cable Routing in Cable Entrance Area.....	30
	RS-422-A .....	32
	RS-232-C .....	34
	Serial port female connector S2 for Debug Interface (RS-232-C) .....	35
	Compact Flash - Card Slot .....	35
<b>7</b>	<b>Membrane Keypad.....</b>	<b>37</b>
	Customer-Specific Keypad .....	38
	Numbering of LEDs .....	38
<b>8</b>	<b>Display .....</b>	<b>39</b>
	Touch Screen .....	39
<b>9</b>	<b>Software.....</b>	<b>40</b>
	Windows CE .....	40
	Generation of Program for Windows CE .....	41

KeTop API Design .....	42
Program for Starting the Application and KeTop API .....	62
SER protocol .....	65
CP001 software interface .....	76
Test tool (VC++ Demo).....	81
zenOn visualization software .....	85
Remote Software ActiveSync .....	108
<b>10 KeTop - Specific Operating Instructions .....</b>	<b>109</b>
Setting of Date and Time .....	109
KeTop Configuration Tool (ConfigTool) .....	110
Checking the Operating and Control Elements .....	115
Installation of Programs .....	116
Saving Files .....	116
<b>11 Options .....</b>	<b>117</b>
Override Potentiometer .....	117
Electronic Handwheel .....	117
Illuminated Push-Button .....	118
Key Switch .....	118
Selector Switch .....	118
<b>12 Accessories .....</b>	<b>119</b>
Wall bracket KeTop WB090 and KeTop WB095 .....	119
Wall Bracket with Height Adjustment plate KeTop WBxxx .....	121
KeTop CB211 Connection Box .....	123
Connection Cable KeTop TTxxx .....	130
Intermediate Cable KeTop ICxxx .....	131
Download Cable KeTop XD040 .....	132
<b>13 Transport Conditions .....</b>	<b>133</b>
<b>14 Technical Data .....</b>	<b>134</b>
<b>15 CE conformity and standards .....</b>	<b>137</b>
CE conformity .....	137
Standards .....	137

## 1 Brief Description

The handheld terminal KeTop T50 is a portable operating and display device with rugged design and Windows-CE compatible electronics. Equipped with a high-performance Intel StrongARM processor and a serial interface, the KeTop T50 is ideal for a great variety of applications:

- Operator panel for machines and plants
- Teach and programming panel for robots
- Test, maintenance and start-up

All tasks can be solved graphically with the large color or b/w display. Operation is intuitive using a touch screen.

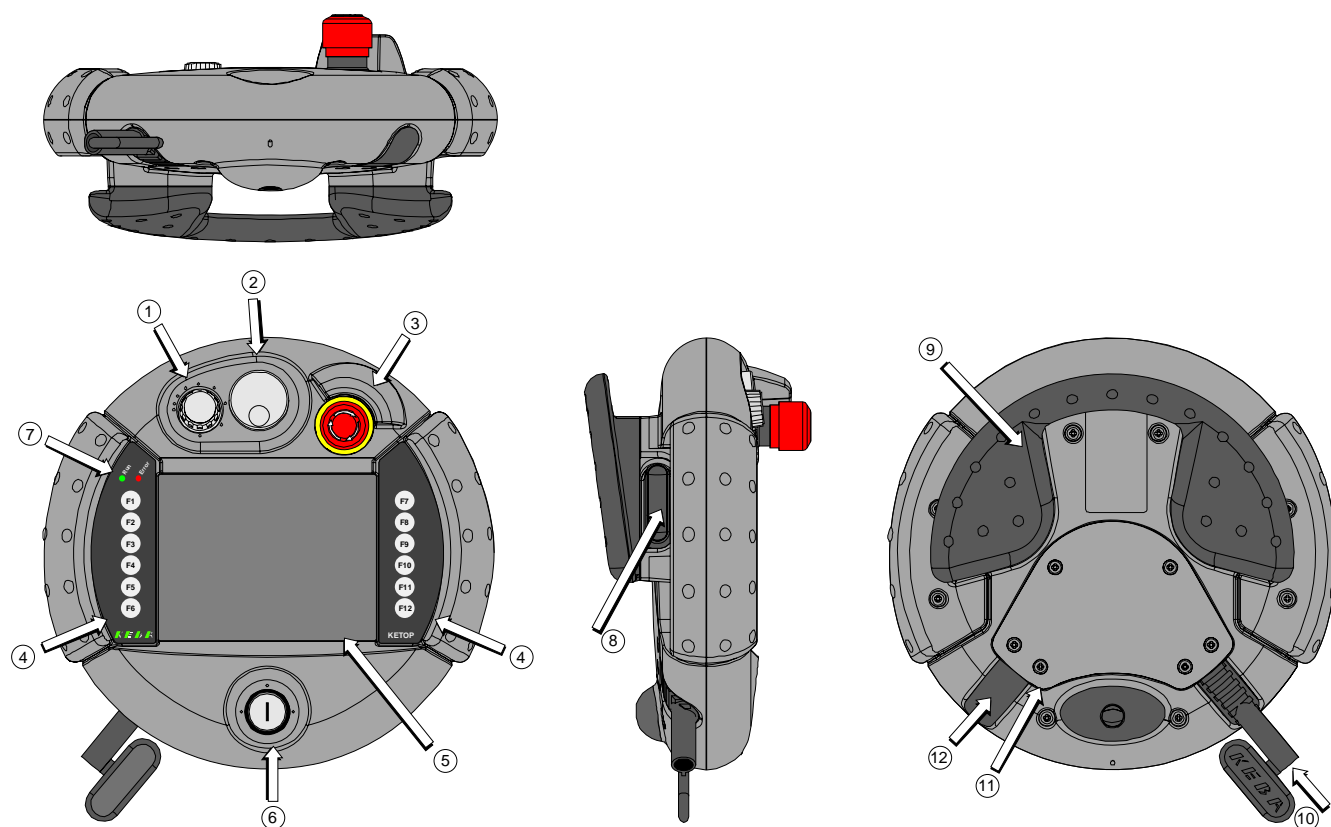
The functionality of the handheld terminal can easily be expanded by using Compact Flash cards of the type I.

The KeTop T50 provides a Windows CE platform for applications generated with common visualization tools or with Visual Basic or Visual C++. Moreover, the KeTop T50 can be connected as a client to a Win NT or Win 2000 server.

Through the use of optional operating and control elements, the KeTop T50 can easily be adapted to the specific application.

## Construction

The KeTop T50 handheld terminal is a portable operating and display device with rugged design and Windows-CE compatible electronics.



- 1 ..... Override potentiometer (option)
- 2 ..... Electronic handwheel (option)
- 3 ..... Emergency stop switch, twin circuit (option)
- 4 ..... Membrane keypad with tactile feedback
- 5 ..... STN LC display with touch screen:
  - b/w 5.7" QVGA resolution 320 x 240 or
  - color 5.7" QVGA resolution 320 x 240 (option)
- 6 ..... 1 location for (option):
  - illuminated push-button, momentary 0
  - illuminated push-button, momentary I
  - illuminated push-button, maintaining I/O,
  - 3-position key switch
- 7 ..... 2 status LEDs

- 8 ..... Two 3-position enabling switches (left and right), twin circuit
- 9 ..... Multigrip handle
- 10 ..... Strain relief for connection cable (delivered with the cable)
- 11 ..... Cable entrance area
- 12 ..... Blind plug for cable outlet not used (to meet protection degree IP65)

Components of KeTop T50



## Ergonomic Features

- Multigrip handle
- Round housing
- Different holding positions
- Operation by right-handed and left-handed people
- Desk top operation
- Operation in wall bracket
- The cable outlet can be on the left or right side of the housing.
- Easy-to-read display

## Housing

- Vibration- and shock resistance
- Housing made of non-flammable material (UL 94-V0), impact-resistant, withstands water, cleaning agents (alcohol and tensides), oil, drilling oils, grease and lubricants
- Twin shell, extremely sturdy housing. Drop-tested on industrial flooring from a height of 1.5 m.

## Operating and Display Panel

- Membrane keys with tactile feedback
- 2 status LEDs
- Buzzer in upper part of housing
- Resistive touch screen, to be operated with finger or stylus
- Backlit STN LC display with
  - b/w 5.7" QVGA resolution with 320 x 240 pixels or
  - color 5.7" QVGA resolution with 320 x 240 pixels

## Hardware

- CPU Intel StrongARM SA-1110/206 MHz
- Memory: SDRAM: 16 MB, FLASH: 16 MB
- Interfaces:
  - Serial interface:  
RS-422-A (option),  
RS-232-C (debug interface in device)
  - Compact Flash card

## 2 Safety Precautions

The device was developed, manufactured, tested and documented in accordance with the applicable safety standards. If you follow the instructions regarding safety and use as described in this manual, the product will, in the normal case, neither cause personal injury nor damage to machinery and equipment.

The instructions contained in this manual must be precisely followed in all circumstances. Failure to do so could result in the creation of potential sources of danger or the disabling of safety features integrated in the handheld terminal.

Apart from the safety instructions given in this manual, the safety precautions and accident prevention measures appropriate to the situation in question must also be observed.

### WARNING

- In all situations where faults occurring on the automation system could cause personal injury or significant damage to machinery and equipment, additional external safety measures must be taken in order to ensure the system as a whole remains in a safe operating condition even in the event of a fault.
- The functionality of the emergency stop switch must be tested regularly.
- The functionality of the emergency stop switch must also be tested after strong shocks (e.g. device falls to ground).
- If the plant is operated with the KeTop take care that, at this time, the operation is only possible with the KeTop and not from any other point of the plant (cf. ANSI/RIA 15.06)
- If the plant is operated with the KeTop in the manual mode (e.g. teaching of robot) take care that the robot moves with slower speed (cf. ANSI/RIA 15.06).

### Notice

- *Measures must be taken to ensure that in the event of power dips or power failures, an interrupted program can be properly restarted.*

## Power Supply

### WARNING

- The device meets the safety class III in accordance with EN61131-2 and EN50178. The 24V power supply for the equipment must be guaranteed through safe isolation of the extra low-voltage circuits from dangerous-contact voltage circuits (e.g. by safety transformers or similar facilities).

### CAUTION

- The power supply circuit must be protected with a fuse of a maximum of 3.15 A

#### *Notice*

- *When planning the power supply, take into account the voltage drop on the KeTop TT xxx connection cable:*

*Specification of power supply lines in the KeTop TT xxx connection cable:*

*Cross section:           AWG24 (0.24mm<sup>2</sup>)  
Material:                zinc-coated copper strand  
Line resistance:        ≤ 90 Ohm/km (≤ 145 Ohm/mile)*

- *The nominal supply voltage directly on the handheld terminal (without KeTop TT xxx connection cable) is:  
+24 VDC (supply voltage range: 18-32 VDC).*
- *Power consumption:  
6 W (333 mA at 18 V DC, 250 mA at 24 V DC)*
- *Maximum interruption time of supply voltage:  
≤ 10 ms (lt. IEC 61131)*

## Emergency Stop Switch

The emergency stop switch of the KeTop meets the requirements of the EN 418. It must be designed as an emergency stop of category 0 or category 1 (see EN 60204-1 chapter 9.2.5.4.2) on the basis of the risk assessment for the machine. The connection of the positive-break contacts to an appropriate monitoring system must meet the safety category which is defined by means of the risk assessment (in accordance with EN 954-1) of the machine.

### WARNING

- As long as the handheld terminal is not plugged to the machine, store the handheld terminal on a place where the operator cannot see it. Take into account that the operator would automatically activate the nearest emergency stop in case of danger. This could have fatal consequences if the emergency stop did not function!
- Emergency stop facilities must remain operational in all operating modes. Resetting an activated emergency stop facility must not result in uncontrolled start-up of machines or installations.
- The emergency stop switch does not replace other safety facilities.
- The emergency stop switch on the handheld terminal does not replace the emergency stop switches to be mounted directly on the machine

## Enabling Switches

Each machine features a normal operating mode and a special operating mode. To guarantee the safety in the normal operating mode, guards and/or operative protection devices which prevent the access are used. The special operating modes are used to maintain the normal mode. In the special mode, the safety must be guaranteed in another way since the operator must enter dangerous areas of the machine, and targeted movements must be possible. In this case, a reduced speed of the machine must be defined by means of the risk assessment. A movement will only be possible if an enabling device is actuated. The user must be trained correspondingly, and he must know the details of the intended use according to the operating manual. The safety-related parts of the control for reducing the speed and for the enabling device must be constructed in a way that they meet the safety category according to EN 954-1, that is defined by means of the risk assessment.

To meet the safety category 3 in accordance with EN 954-1:1996, the enabling switch must be realized with 2 circuits.

The EN 60204 describes the functioning of the enabling device. Due to the latest findings of analyses of accidents and since technical solutions are

available, the 3-position enabling switch became the state of the art. The positions 1 and 3 of the enabling switch are OFF functions. Only the central position is used for enabling. The EN 60204-1:1997 is identical with the IEC 60204-1. So the 3-position enabling switch is of international relevance.

Further the EN 60204-1:1997 prescribes that the enabling switches shall be connected to a category 0 stop or a category 1 stop.

## WARNING

- The enabling switch is only suitable as safety function if the operator activating the enabling switch recognizes the dangerous situation in time so that he can immediately take the necessary measures to avoid such situations. As additional measure reduced speed of the movement can be necessary. The allowed speed must be determined by means of a risk assessment.
- The enabling switch is only used to enable commands for performing dangerous movements. The commands themselves must be activated by a separate operating element (key on handheld terminal). Only the person who operates the enabling switch is allowed to work in the dangerous area.

### *Notice*

- *At the KeTop, the enabling switches always feature two circuits.*

## Risk Assessment of Machinery

For the risk assessment the following standards must be applied:

- EN 292 „General principles for design of machinery“
- EN 1050 „Principles for risk assessment of machinery“
- EN 954-1 „Safety-related parts of control systems“
- ANSI/RIA 15.06-1999 „For Industrial Robots and Robot Systems-Safety Requirements (Section 9)“

The safety categories (B, 1, 2, 3, 4) which finally affect the design of the safety-related parts of a machine result from this risk assessment.

The connections with two different control relays on page 21 and 23 show how the **safety category 3** can be fulfilled with the KeTop and its safety-related parts. The entire concept of the machine must be laid out according to the principles of safety category 3.

## 3 General Instructions

### Intended Use of KeTop

The KeTop may only be used for the purposes mentioned in this manual (also refer to chapter "Brief Description" on page 5).

### Waste disposal

Observe the national regulations when disposing of electronic components!

### Handling of KeTop

You have chosen a high-quality KeTop that is equipped with highly sensitive state-of-the-art electronics.

To avoid malfunctions or damage through improper handling, follow these instructions during operation.

#### CAUTION

- Turn off the power supply before opening the cable entrance area of the KeTop. Otherwise the components could be destroyed or undefined signals could occur.
- Make sure that nobody can fall over the cable to avoid that the device falls to ground.
- Take care not to squeeze and thus damage the cable with any object.
- Do not lay the cable over sharp edges to avoid damaging the cable sheath.
- If you do not use the device, hang it into the wall bracket KeTop WBxxx provided for storage.
- Do not lay down the device with the operating side facing down to avoid damaging the operating elements.
- Never lay the device onto unstable surfaces. It could fall to ground and thus be damaged.
- Never lay the device close to heat sources or into direct sunlight.
- Avoid exposing the device to mechanical vibrations, excessive dust, humidity or to strong magnetic fields.
- Never clean the device, operating panel and operating elements with solvents, scouring agent or scrubbing sponges. For cleaning the device, use a soft cloth and a bit of water or a mild cleaning agent.



- Make sure that no foreign objects or liquids can penetrate into the device. Check at regular intervals the protective covers of the device, if all housing screws are firmly tightened and if the housing or the cable entry is damaged.
- If the device shows any defect, please send it, including a detailed error description, to your supplier or the relevant after-sales service office.



When the cable entrance area is open, the KeTop is sensitive to electrostatic discharge.

## CAUTION

- For operating the touch-screen use your finger or a touch-pen. Never use sharp objects (e.g. screwdriver,...) for operating the touch screen. This could damage the touch screen.

## Foreseeable Misuse of Enabling Switch

Foreseeable misuse means the not allowed fixing of the enabling switch in the enabling position. The foreseeable misuse of the enabling switch must be restricted. The following measures causing the stop of the machine in the manual mode are recommended:

- Inquiry of the enabling switch when turning on the machine/plant and inquiry of the enabling switch when changing the operating mode from automatic to manual (The enabling switch must not be in the enabling position.).
- The enabling switch must be released within a defined period and pushed into the enabling position again. The length of the period must be defined according to the activity.

## 4 Safety Elements

### Enabling Switch

The KeTop is equipped with two enabling switches, one at the left and one at the right side of the device. This allows a left- and right-hand operation of the enabling switch. Both enabling switches are equivalent and parallel switched. So for enabling only one of both enabling switches must be activated.

The electronic enabling switch realizes the enabling equipment as safety function for machines in special operating modes. The enabling switch is only a part of this equipment.

The enabling switch consists of a 3-position operating element and a separated evaluation electronics. An essential feature are the continuous two-channel circuits beginning from the actuating elements up to the connecting terminals. For the evaluation circuits different technologies and circuits are used. Due to the electronic switching contacts, their lifetime does not depend on the load provided the nominal values of the load (ohmic, inductive and capacitive) are not exceeded.

### Functioning

The actuating element consists of two symmetrically arranged slides. The position of these slides is detected by electrical switches and transmitted to the evaluation electronics.

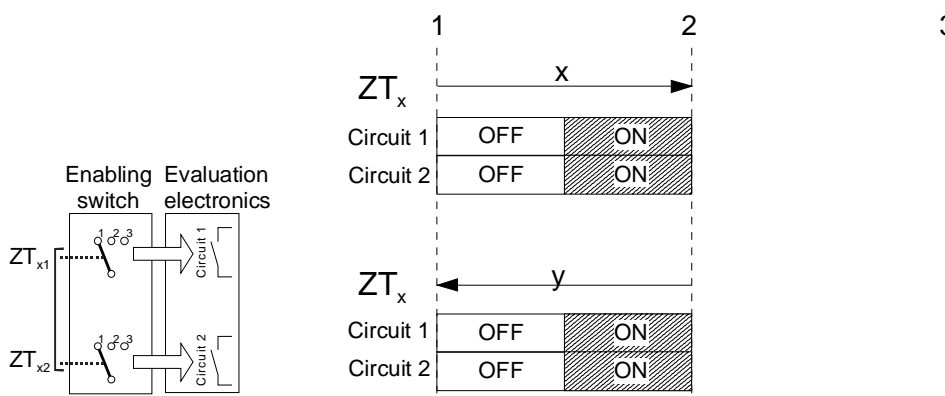
Positions of enabling switch:

Position	Function	Enabling switch	Contacts
1	home position	is not pressed	enabling outputs are open
2	enabling	is pressed	enabling outputs are closed
3	panic	is pressed strong	enabling outputs are open

For the enabling switch, the following switching sequences are possible:

## Enabling

Home position  $\boxed{x}$  enabling  $\boxed{y}$  home position



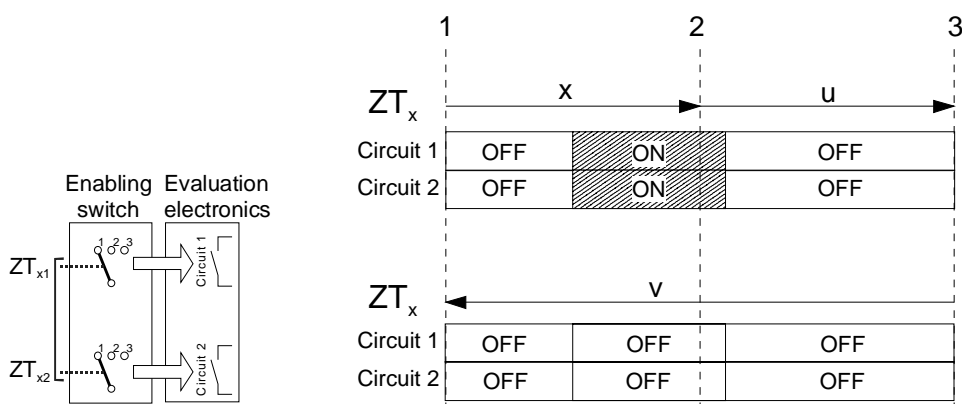
x ... L (Left) or R (Right)

Contact travel diagram for enabling

## Panic

The pushing of the actuating elements into the panic position is evaluated in a way that the enabling position is skipped when the actuating elements are released.

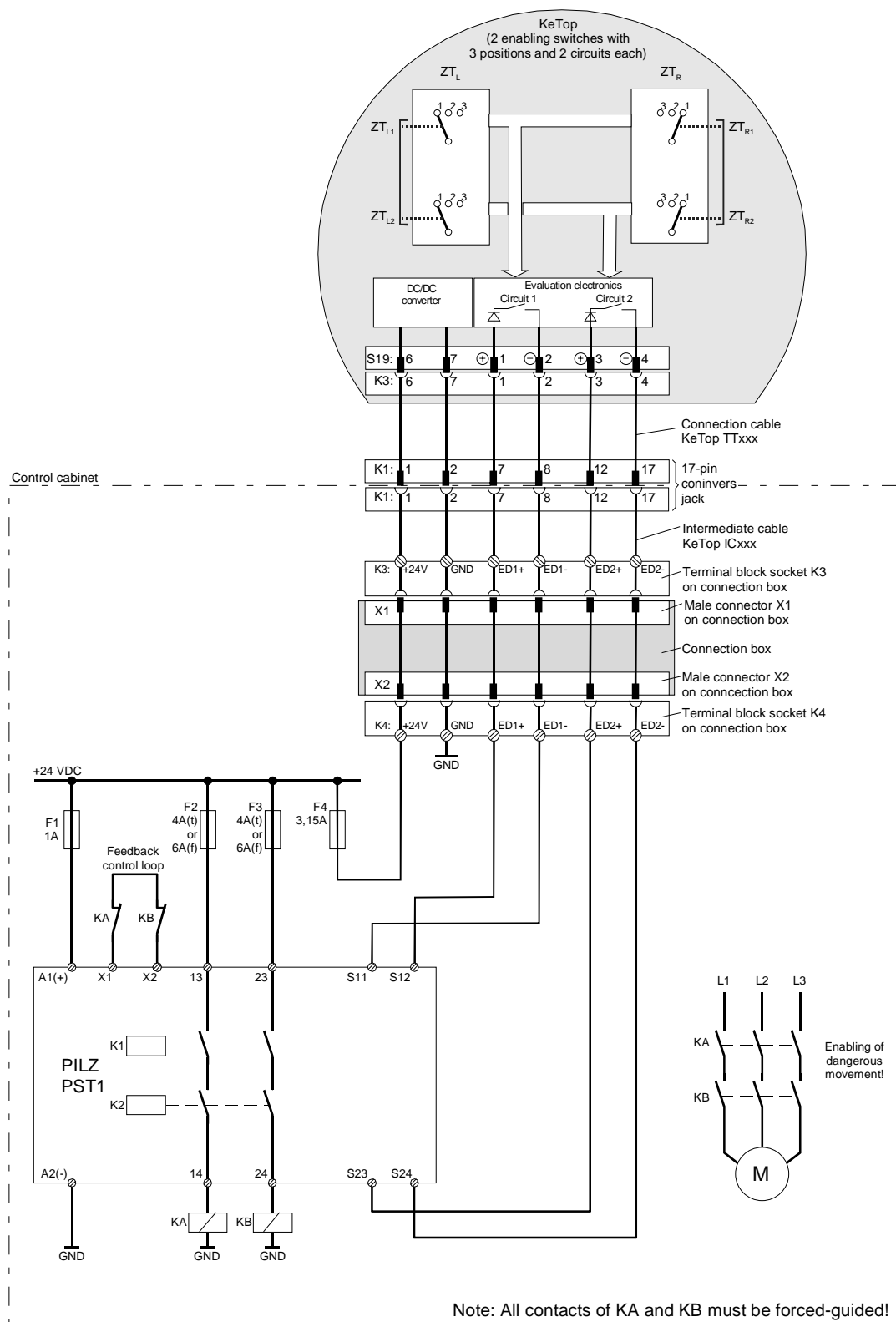
Home position  $\boxed{x}$  enabling  $\boxed{u}$  panic  $\boxed{v}$  home position



x ... L (Left) or R (Right)

Contact travel diagram for panic

### Example of Connection with PILZ PST1 Control Relay



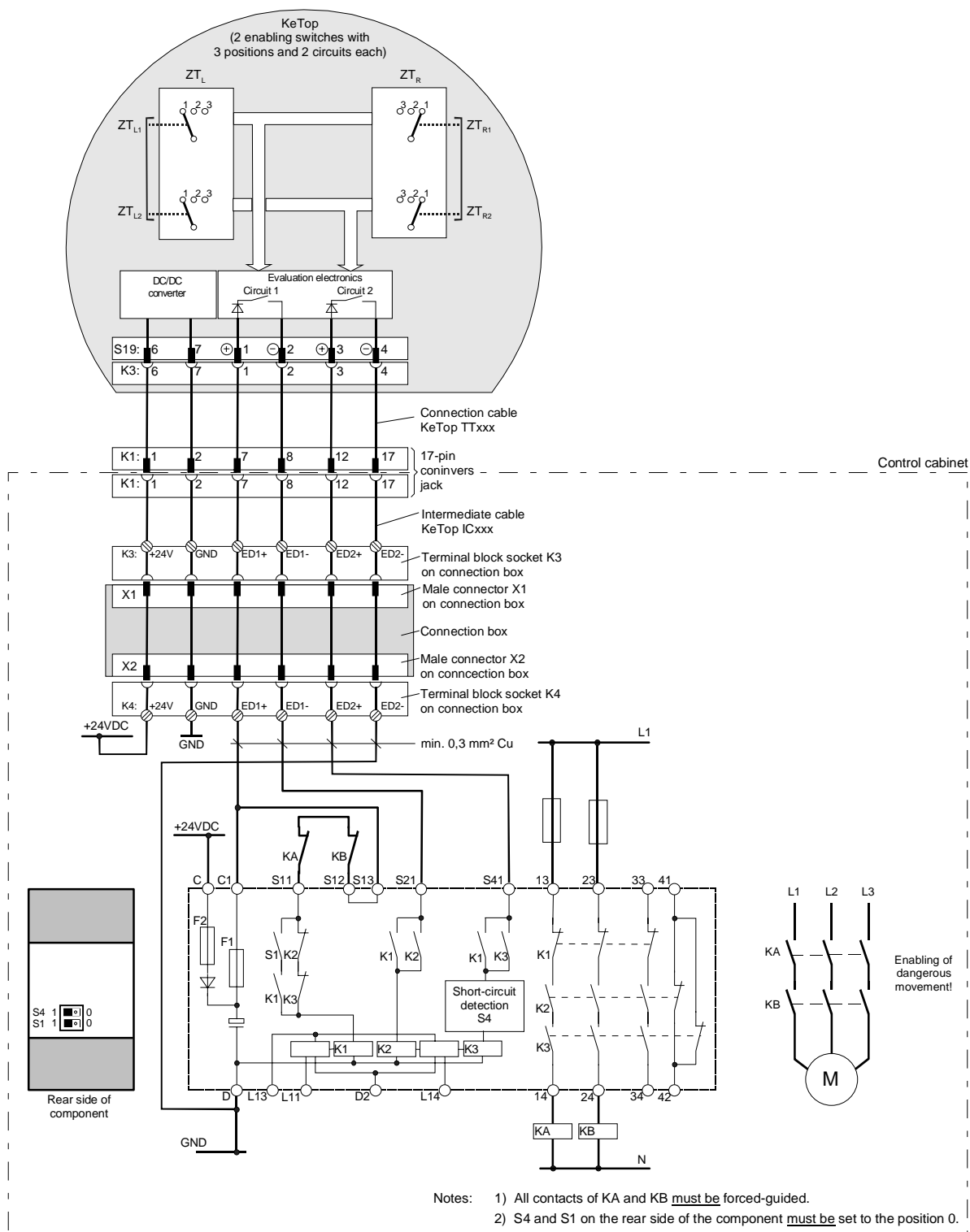
Suggested wiring of enabling switches to fulfill safety category 3 with PILZ control relay. Also follow the instructions described in the PILZ operating manual about the PST1 device.

**Functional procedure:**

- Only if both channels are activated „simultaneously“ (by pressing one of the enabling switches) both output relays K1 and K2 will energize and the output contacts 13-14 and 23-24 will close.
- The output relays K1 and K2 will not energize if
  - only one enabling channel is activated (in case of a defect),
  - the tolerance value for the simultaneity period is exceeded,
  - the feedback control loop X1-X2 is open.
- If one enabling channel is deactivated after being simultaneously activated (by releasing the enabling switch or in case of a defect), the output relays K1 and K2 will return to their initial position again. The forced-guided output contacts 13-14 and 23-24 will open. The output relays will energize again only after both enabling channels have been deactivated and simultaneously activated once again.

In this way the enabling switches avoid that one single error makes the safety function inoperational. A single error will be recognized at the next cycle at the latest.

# Example of Connection with ELAN SRB-NA-R-C.27/S1 Control Relay



Suggested wiring of enabling switches for safety category 3 with the ELAN SRB-NA-R-C.27/S1 control relay. In addition follow the instructions of the operating manual about the SRB-NA-R-C.27/S1.

**Functional procedure:**

- Only if both channels are activated „simultaneously“ (by pressing one of the enabling switches) both output relays K2 and K3 will energize and the output contacts 13-14, 23-24 and 33-34 will close.
- The output relays K2 and K3 will not energize if
  - only one enabling channel is activated (in case of a defect),
  - the tolerance value for the simultaneity period is exceeded,
  - the feedback control loop S11-S12 is open.
- If one enabling channel is deactivated after being simultaneously activated (by releasing the enabling switch or in case of a defect), the output relays K2 and K3 will return to their initial position again. The forced-guided output contacts 13-14 and 23-24 will open. The output relays will energize again only after both enabling channels have been deactivated and simultaneously activated once again.

In this way the enabling switches avoid that one single error makes the safety function inoperational. A single error will be recognized at the next cycle at the latest.

**Technical Data of Switching Elements of Enabling Switches**

Nominal voltage	24 V DC (typ.) 32 V DC (max.)
Nominal current	500 mA (typ.)
Short-circuit current	circuit 1: max. 1,9 A circuit 2: max. 600 mA
Max. inductive load (at 500 mA)	circuit 1: max. 1H circuit 2: max. 320 mH
Max. capacitive load	circuit 1: no limit since the transistor is protected thermally circuit 2: max. 500 µF

The switching elements of the enabling switches are protected against reversed polarity. The outputs of both circuits are protected against short circuits and excess load.

Circuit 1: thermal protective circuit

Circuit 2: fold back line



## Emergency Stop Switch

The emergency stop switch used on the KeTop features two circuits. The contacts are normally closed.

### Connection values

- Connection voltage: 24 VDC
- Maximum current: 500 mA
- Minimum current: 10 mA

## 5 Shielding

The KeTop is conceived for industrial applications. In this environment, considerable electromagnetic and electrostatic interferences may occur. For this reason, particular importance was attached to immunity to interferences and data security when the device was constructed.

- All shielding and filtering measures (filtered supply voltage) are directly realized in the KeTop.
- Due to its special construction, the KeTop cable ensures immunity to interferences in case of larger cable lengths.  
I.e. the *data lines* (communication signals) are shielded and routed separately from the *control lines* (power supply, enabling switch, E-stop, key switch, etc.) in the KeTop cable.
- Normally the power supply lines in the control cabinet are not shielded. Therefore they are routed outside the KeTop cable shield to avoid that the sensitive data lines are disturbed by the other lines (e.g. power supply lines).

### Shield Connections

The cable shield of the KeTop cable can be considered as extension of the KeTop shield housing (=PCB) to the shield housing of the communication partner (e.g.: PLC). In this way the shield connections of the cable shield to the shields of the devices essentially contribute to the immunity to interferences of the KeTop. Connections to ground are not required for shield connections.

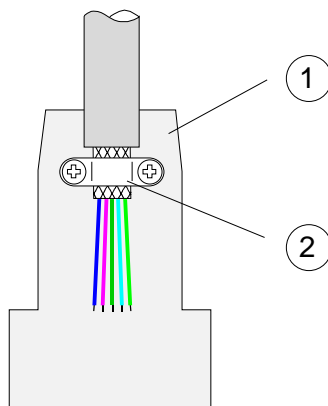
The shield connection in the KeTop is made via the RJ-45 connector in the cable entrance area.

The connection of the KeTop via the connection box (e.g. CB211) guarantees a reliable shield connection up to the communication partner.

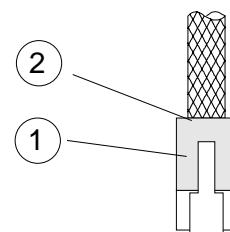
All connection cables available at KEBA (KeTop TTxxx, KeTop ICxxx, KeTop XD040, KeTop CD040) guarantee a proper shield connection.

At the cables you produced yourself, the following conditions for shielding the data lines must be fulfilled:

- Provide a large contact surface area at each connector (DSUB or RJ-45) when connecting the cable shield to the connector housing.



DSUB connector



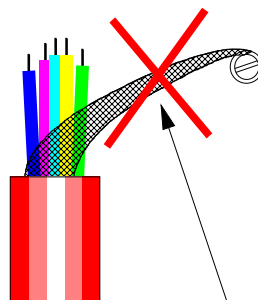
RJ-45 connector

- 1..... Metallized or metallic housing  
 2..... Provide large contact surface area.

---

Correct shield connection at DSUB and RJ-45 connector

- Pig tails to connect the shield to the pins or terminals are not suitable. The inductivity of such pig tails represents a high-ohmic resistance for high-frequency interferences. This seems to interrupt the cable shield. In this case interferences are not absorbed any more but directly affect the inner lines.



No pig tails

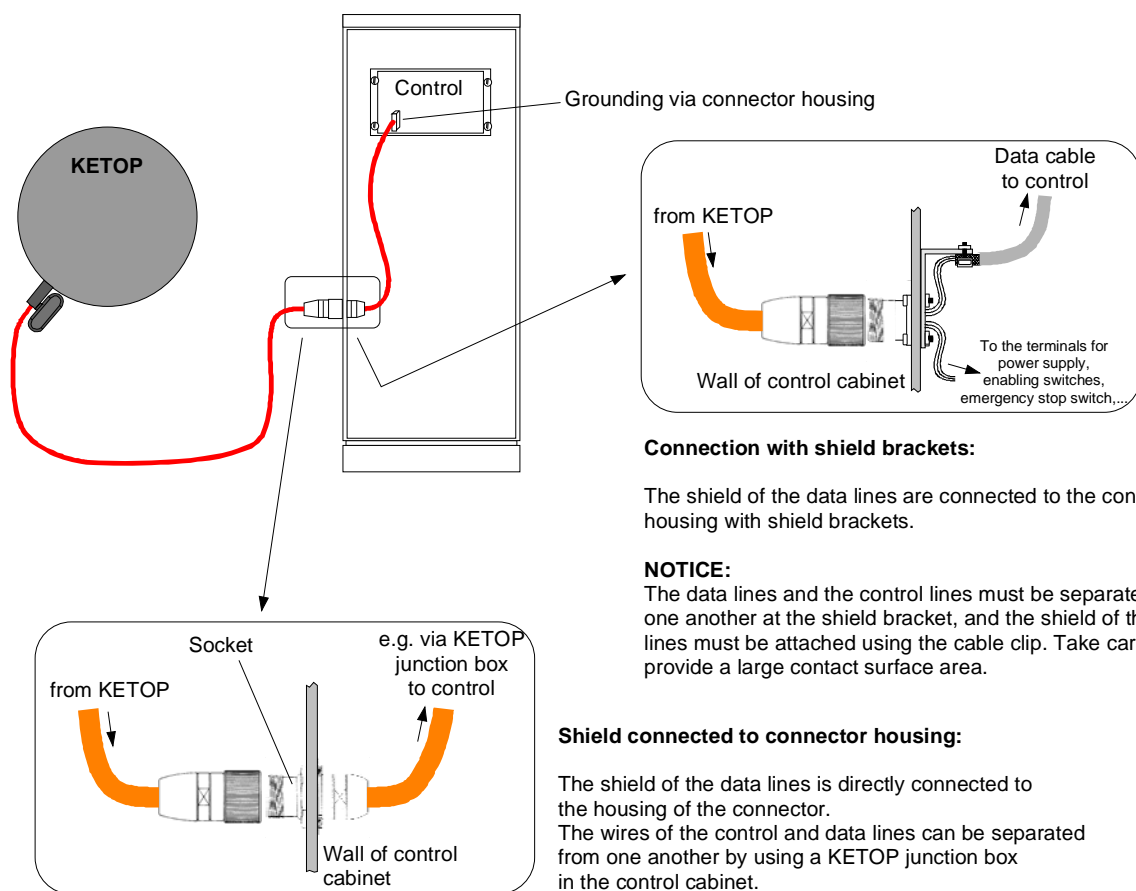
---

Insufficient connection of cable shield

## Shielding inside the Control Cabinet

In many cases, several interference sources exist in the control cabinet, such as servo drive modules, transformers, contactors and relays. Therefore it is necessary to continue the cable shield from the connector shell (control cabinet) up to the PLC (continuous connection from handheld terminal up to PLC).

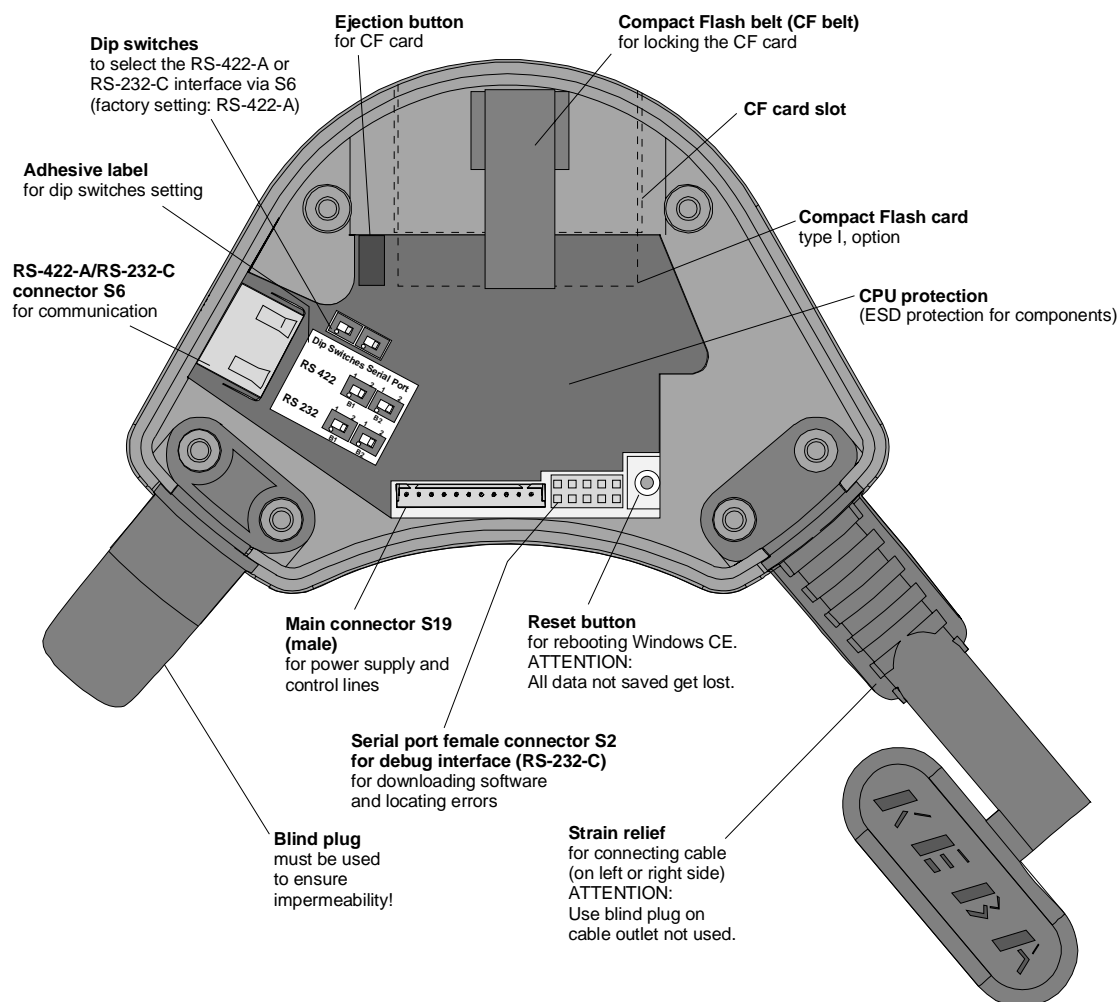
This continuous connection is realised through the connection of the connector shell to the flange socket of the control cabinet, and inside the control cabinet through the connection of the cable shield to the control cabinet (using cable clips). The smaller the distance between connection of connector shell on the control cabinet and the cable clip of the control cabinet is, the better the shield damping will be.



Connection of cable shield in a control cabinet

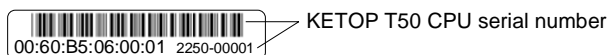
## 6 Connection

### Cable Entrance Area



Cable entrance area of KeTop T50

### Adhesive label for CPU serial number



Adhesive label for CPU serial number

## Cable Routing in Cable Entrance Area

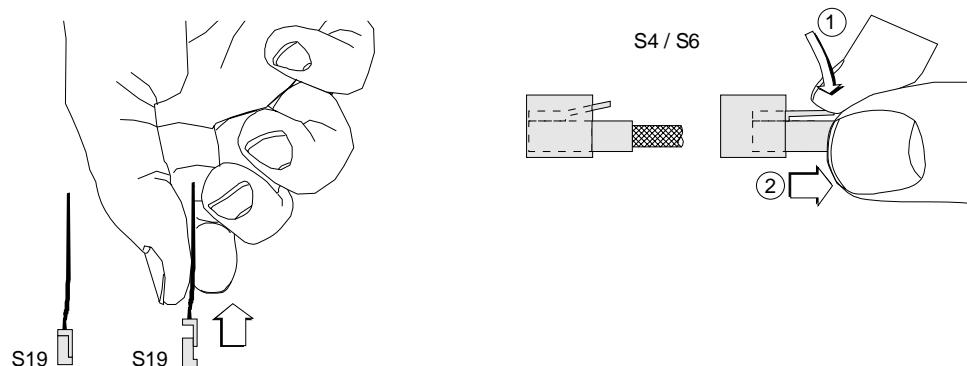
After opening the cable entrance area, the connecting lines can be routed as described in the following chapters. Before opening the KeTop please pay attention to the following safety instructions:

### *Instructions for opening the cable entrance area:*

- Lay the KeTop with the display facing down onto a plane and clean table (preferable on ESD pad) and take care not to damage the KeTop and its operating elements.
- For opening and closing the cable entrance area use the following type of screwdriver: „Phillips size 2“.

### *Instructions for modifications in the cable entrance area:*

- Unplug the main connector (S19) by pulling on its wires with your fingers. Do not use any sharp objects.
- For unplugging the RJ-45 jack (S4 / S6), actuate the locking lever:



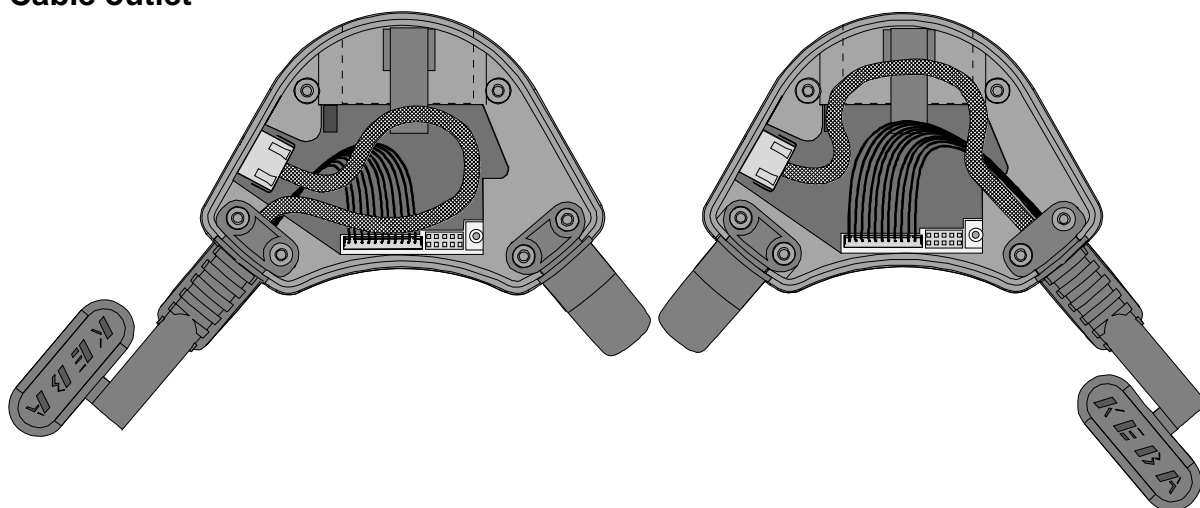
## **⚠ WARNING**

- Make sure that the connectors S19 and S6 correctly snap in when you plug them in. Otherwise the emergency stop functionality (S19) or the correct shielding (S4/S6) might not be given any more.

*Instructions for closing the cable entrance area: Make sure that*

- *the sealing is clean, not damaged and correctly positioned in the cable entrance area,*
- *no cables are squeezed in,*
- *the cover of the cable entrance area is attached again with all 6 screws (torque: 0.4 bis 0.5 Nm). Otherwise the protection degree cannot be guaranteed.*

## Cable outlet



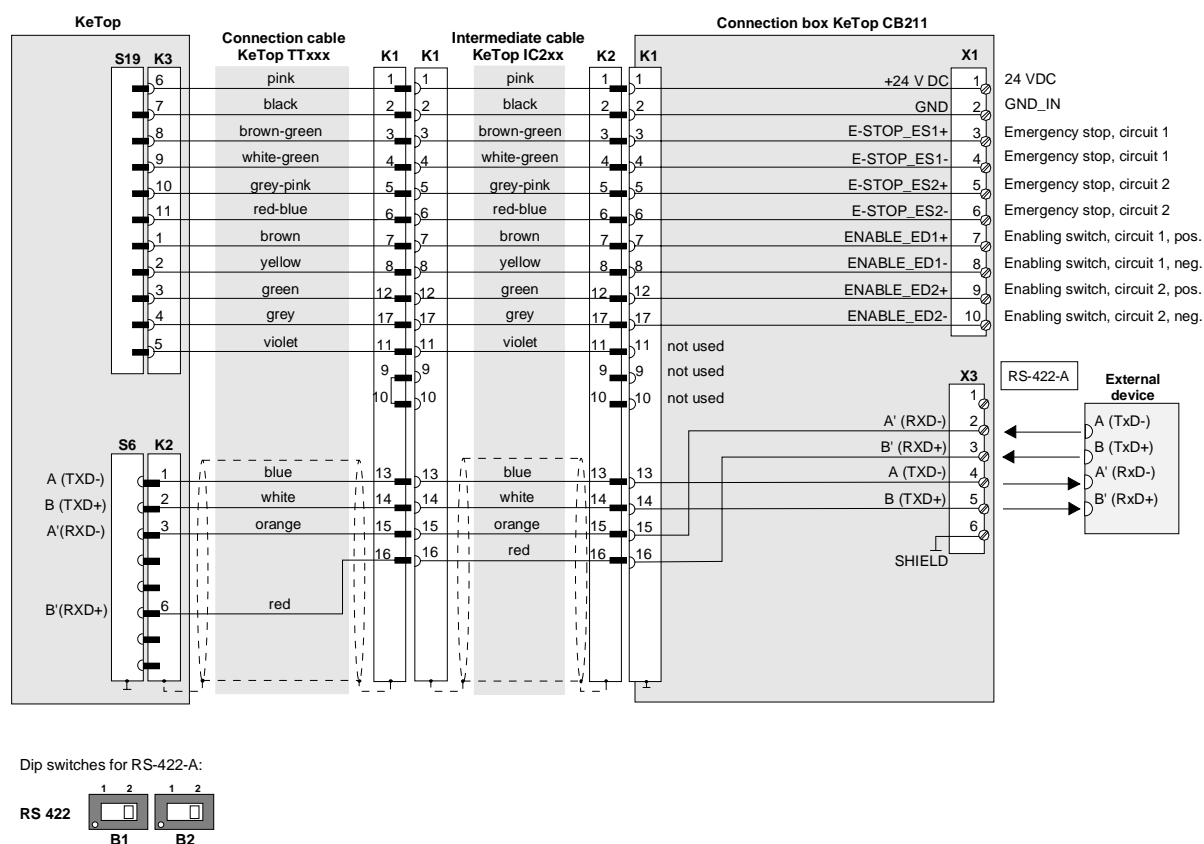
Cable outlet on left and right sight with data and control lines

## RS-422-A

The communication takes place via the COM module connector S6 in the cable entrance area of the KeTop.

The RS-422-A interface assigned to the COM 5 interface port is in the software. The interface parameters are set via the WIN32API in the Windows operating system.

## Connection diagram



Connection diagram: KeTop with RS-422-A via connection box CB211



**General information about the RS-422-A interface**

- The A terminal of the generator shall be negative with respect to the B terminal for a binary 1 (MARK or OFF) state.
- The A terminal of the generator shall be positive with respect to the B terminal for a binary 0 (SPACE or ON) state.

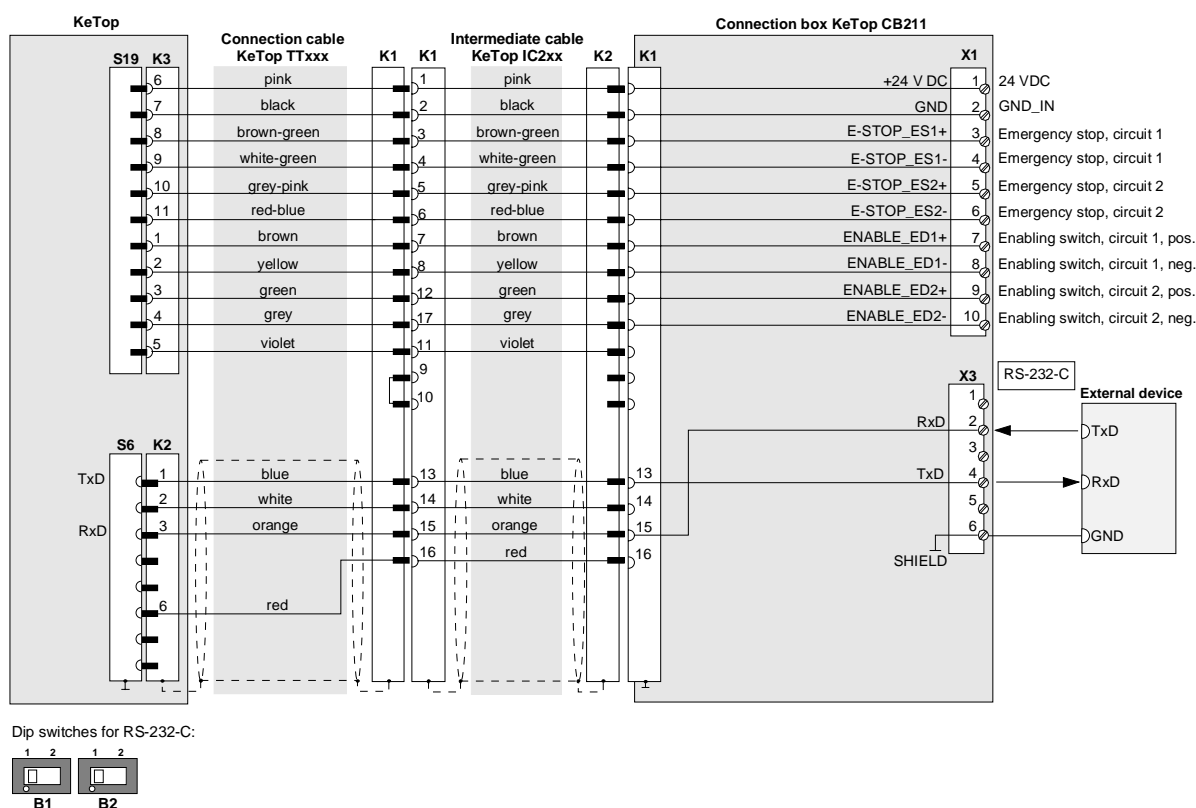
To identify the lines, the voltage between the lines A and B can be measured by means of a voltmeter.

## RS-232-C

The communication takes place via the COM module connector S6 in the cable entrance area of the KeTop.

The RS-422-A interface assigned to the COM 2 interface port is in the software.

## Connection diagram



Connection diagram: KeTop with RS-232-C via connection box CB211

## Serial port female connector S2 for Debug Interface (RS-232-C)

Using the „Boot-Loader“ software, the „serial port“ interface can be used for debugging and for downloading software. Using the remote software ActiveSync, it can be used for adjusting and transmitting data from and to a PC.

For that purpose, the download cable KeTop XD040 is available.

The following interface parameters are defined and cannot be changed:

- 38400 baud
- 8 data bits
- 1 stop bit
- No parity
- No handshake

The debug interface is assigned to the COM 1 interface port in the software.

## Compact Flash - Card Slot

The Compact Flash card slot may be used for Compact Flashes of the type 1.

The following list shows the CF cards that have been tested by KEBA. The following CF cards can be recommended for the use in the KeTop:


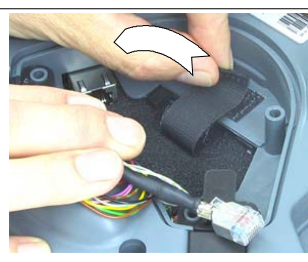
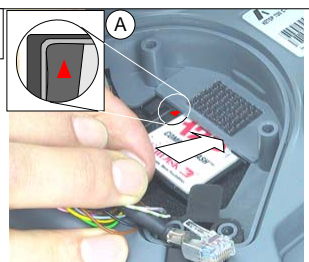
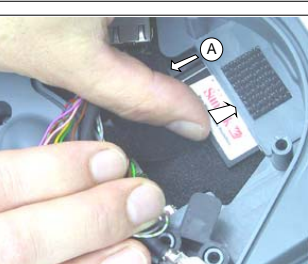


**Type: Smart Modula**

Memory sizes: 32 , 64 and 128 MB







**Type: Sandisk**

Memory sizes: 32, 64 and 128 MB

## Inserting the Compact Flash (CF) Card

<p>1</p>  <p>Lay the KETOP with the display facing down onto a plane and clean table (preferably on ESD pad) and take care not to damage the KETOP and its operating elements.</p>	<p>2</p>  <p>Open the cable entrance area by releasing the screws, unplug the connector S6 and open the CF belt.</p>	<p>3</p>  <p>Insert the CF card into the CF slot in the direction of the arrow (A).</p>
<p>4</p>  <p>Insert the CF card until it locks in and the ejection button (A) jumps out.</p>	<p>5</p>  <p>Fasten the CF card with the CF belt to prevent it from falling out in case of impacts.</p>	<p>6</p>  <p>Make sure that the cover of the cable entrance area is attached again with all 6 screws (torque: 0.4 bis 0.5Nm) otherwise the protection degree IP54 cannot be guaranteed.</p>

## Removing the Compact Flash (CF) Card

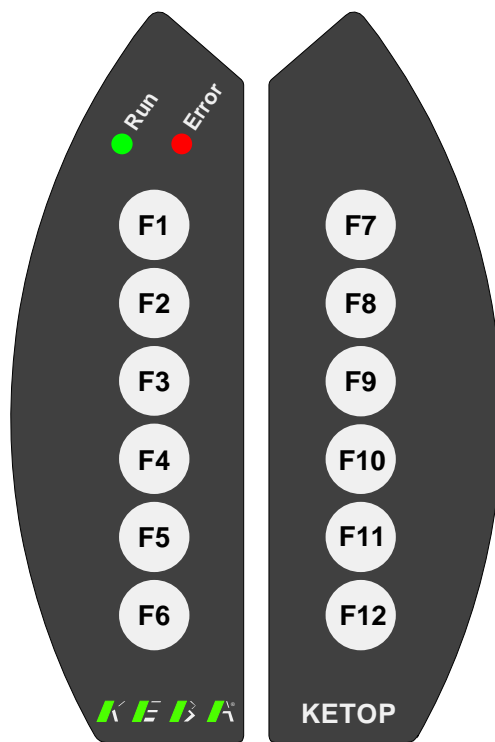
<p>1</p>  <p>Lay the KETOP with the display facing down onto a plane and clean table (preferably on ESD pad) and take care not to damage the KETOP and its operating elements.</p>	<p>2</p>  <p>Open the cable entrance area by releasing the screws, unplug the connector S6 and open the CF belt.</p>	<p>3</p>  <p>Press the ejection button. Attention: Take care not to press on to the CF card when pressing the ejection button.</p>
<p>4</p>  <p>The CF card is ejected and can be taken out.</p>	<p>5</p>  <p>Close the CF belt.</p>	<p>6</p>  <p>Make sure that the cover of the cable entrance area is attached again with all 6 screws (torque: 0.4 bis 0.5Nm) otherwise the protection degree IP54 cannot be guaranteed.</p>

## 7 Membrane Keypad

### General Membrane Keypad 5.7 (320 x 240)

The keypad assignment is stored in the registry and defined for the general membrane keypad as follows:

Key Labelling	Windows Virtual Key Code	Hex
F1	VK_F1	0x70
F2	VK_F2	0x71
F3	VK_F3	0x72
F4	VK_F4	0x73
F5	VK_F5	0x74
F6	VK_F6	0x75
F7	VK_F7	0x76
F8	VK_F8	0x77
F9	VK_F9	0x78
F10	VK_F10	0x79
F11	VK_F11	0x7A
F12	VK_F12	0x7B



General membrane keypad 5.7" display

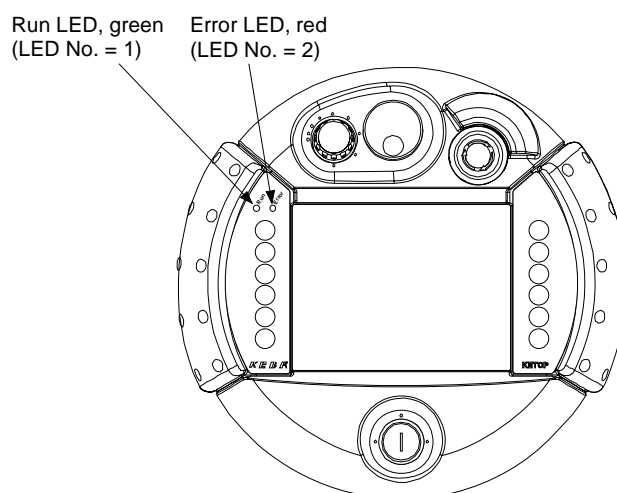
## Customer-Specific Keypad

The customer-specific labeling and layout of the KeTop keypad is possible as well.

## Numbering of LEDs

This chapter describes the numbering of the keypad LEDs. The LED number is needed for activating the LED.

A LED is activated via the KeTop API design (see chapter „KeTop API Design -> KtpSetKeyboardLed“ on page 55)



Numbering of LEDs on the KeTop

## 8 Display

The KeTop is available with the following two displays:

	<b>Color display</b>	<b>b/w display</b>
Type:	graphics-capable color STN LCD	graphics-capable b/w STN LCD
Size:	5.7" (120 x 90 mm)	5.7" (120 x 90 mm)
Resolution:	1/4 VGA, 320 x 240 pixels	1/4 VGA, 320 x 240 pixels
Representation:	256 colors	16 grey scales
Background lighting:	CCFT cold cathode tube	CCFT cold cathode tube
Touch screen:	analog-resistive	analog-resistive

The lifetime of the background lighting (min. 50,000 h) can be increased by activating a screensaver via the Control Panel. Procedure:

*Start -> Settings -> Control Panel -> Display -> Folder Backlight  
Select the function „Automatically turn off backlight while on external power“*

Here you can enter the turn-off time (30s to 30min).

## Touch Screen

If the touch screen is not operated with the finger, we recommend using a touch stylus, for example touch styluses of PDAs by PALM or SONY.

The touch screen is already calibrated when the KeTop is delivered. No further calibration is required.

If a re-calibration is required for any reason (humidity of air, temperature, etc.), perform the calibration procedure under Windows CE as follows:

*Start -> Settings -> Control Panel -> Stylus -> Calibration -> Recalibrate*

For the calibration you can also use the ConifgTool. See chapter „Display and Touch Screen Settings“ on page 111.

## 9 Software

### Windows CE

The KeTop T50 is delivered with pre-installed software that is stored in the flash of the device. This software is also called "Image" since it combines all software contents in one file. The image contains the Windows CE operating system core and possibly additional applications. The image is created by means of the MS Platformbuilder for Windows CE.

### Software of Standard KeTop T50

#### Operating system Windows CE 3.0:

- System control
- Command prompt
- Windows Explorer

#### Flash file system:

- IPSM (Intel Persistent Storage Manager)

#### Pre-installed application programs:

- Pocket Notepad
- Pocket Registry Editor
- Active Sync

#### KeTop-specific software:

- KeTop Configuration Tool
- Registry Backup
- Set Time, temporary time and date entry
- TouchClean, Touch Cleaning Application
- Version, overview of versions
- Toggle-Sip (software keyboard)
- KeTop tools (extension in control panel for configuration of contrast, background lighting, screensaver, touch screen and joystick calibration, onboard Ethernet on/off, clearing of registry, image update, ..)

#### Test tools:

- TestSerial, Testprogramm für serielle Schnittstellen  
(\windows\testserial.exe)



## Generation of Program for Windows CE

The user can easily generate programs for Windows CE. Programming is similar to an application for a standard MS Windows NT PC. Under Windows CE, only the number of available WIN32-APIs is limited.

### Prerequisites for Application Programming

- **MS Windows CE eMbedded Visual Tools 3.0**, already contains the complete programming environment (MS Visual C++ 6.0, MS Visual Basic 6.0, MS Windows CE Toolkit, SDK for HPC / HPC Pro)
- If special hardware-dependent functions of the KeTop should be used, the SDK of the KeTop must be installed subsequently since the delivered SDK for HPC (Handheld PC) and HPC Pro (Handheld PC Professional) does not take into account certain customer-specific adaptations.
- **SDK** (Software Development Kit) by manufacturer of Windows CE device (KEBA).  
The SDK provides to the programming environment the processor-dependent header and library files of the OEM adaptation.

## KeTop API Design

The entire SDK is implemented in a single dynamic link library (DLL). All functions described in this document are exported from this DLL. For Visual Basic, the module file KeTopAPI.bas must be imported into the Visual Basic project. This file and the corresponding SDK are contained on the CD „KeTop SK001“.

The following platform names are assigned to the KeTops:

KeTop T100: M750\_100\_1

KeTop T50: 50\_1

## Notes on Programming under Visual Basic

- For the remote connection between „Visual Studio“ and the KeTop T100 respectively T50, the PPP transport protocol must be used. Set the PPP transport protocol as follows:
  - ▶ Tools -> Remote Tools -> „*Configure Platform Manager...*“
  - ▶ A window appears. Click on the entry “M750\_100\_1” for the platform of the KeTop T100 respectively on the entry “50\_1” for the platform of the KeTop T50.
  - ▶ Click on “+” to open the lower level “M750\_100\_1(Default Device)” respectively “50\_1(Default Device)” and select “Properties”.
  - ▶ Select under “Properties” the transport protocol “PPP Transport for Windows CE”.
- System files that are copied to \windows directory by Visual Studio remain stored permanently. For this reason, these files must be copied manually to the directory \IPSM\windows (must be created by the user if necessary).

In case of a restart of Windows CE, these files are automatically copied to the \windows directory. Therefore these files will be available for the operating system and after a restart (valid for OEM Build version 1.4 and higher).

The Visual Basic system files are contained on the CD „KeTop SK001“ from where they can be copied to the directory \IPSM\windows of the KeTop T100.

To store the project file \*.vb permanently on the KeTop T100 respectively T50, the remote path must be changed correspondingly in the Visual Studio. Procedure:

- ▶ Project -> *[project name] Properties...*
- ▶ A window appears. Select the tab "General".
- ▶ Enter the remote path for the project file.  
*Notice: The remote path must refer to the directory \IPSM or to a corresponding lower directory.*

## Common data types

This section contains a detailed description of the common data types for the communication with the handheld terminal. For further information, see TpuHwDataTypes.h.

INT8	Signed 8 bit integer variable.
UINT8	Unsigned 8 bit integer variable.
BacklightStat	Enum, displays the backlight status
JoystickPos	Struct, for joystick data.
Status Struct	Describes the startup state of the device.
EventMsg	Enum, describes the event message received.
EventMsgDomains	Enum, describes the events a handler has been subscribed to.
eventCallback	Function pointer to callback function.

## Handling of Errors

### Rules

- All functions expecting any input parameter check if the parameter is located inside the range and if its data type is correct. If a parameter is located outside the range the function will return INVALID\_ARG\_RANGE.
- All functions expecting a pointer for output data as parameter check if the pointer is valid, i.e. the pointer must not be NULL. If the pointer is invalid, the function will return INVALID\_ARG\_INVALID\_PTR.
- All functions having any string as parameter check if the pointer to the string is valid. If the pointer is invalid, the function will return INVALID\_ARG\_INVALID\_STR\_PTR.

### Defines

SUCCESS	0
OK	0
FAIL	1
INVALID_ARG_RANGE	2
INVALID_ARG_PTR	3
INVALID_ARG_STR_PTR	4
INVALID_ARG_UNKNWN_COOKIE	5
INVALID_ARG_UNKNWN_DOMAIN	6
INVALID_NOT_CALIBRATED	7

## Functions for starting and closing

This chapter describes functions that are needed for starting and closing the KetopApi.dll.

### KtpAPIInit

- Declaration:  
Visual C:       UINT8 KtpAPIInit(void);  
Visual Basic:   KtpAPIInit() As Byte;
- Description:  
This method initializes the KetopAPI.

### KtpAPIDeinit

- Declaration:  
Visual C:       void KtpAPIDeinit(void);  
Visual Basic:   KtpAPIDeinit()
- Description:  
This method cancels all initializations of the KtpAPIInit.

## Functions for Configuration

This section describes the functions that are available for the configuration of the device. All functions return one of the above defines as error code.

### KtpSetBrightness

- Declaration:  
Visual C:     UINT8 KtpSetBrightness(/\*[in]\*/UINT8 u8\_Brightness);  
Visual Basic:   KtpSetBrightness(ByVal brightness As Byte) As Byte
- Description:  
This method sets the brightness of the LC display on the device.
- Arguments:  
0-7, 0 = min, 7 = max

### KtpSetContrast

- Declaration:  
Visual C:     UINT8 KtpSetContrast(/\*[in]\*/UINT8 u8\_Contrast);  
Visual Basic:   KtpSetContrast(ByVal contrast As Byte) As Byte
- Description:  
This method sets the contrast of the LC display on the device.
- Arguments:  
0-31, 0 = min, 31 = max  
(KeTop T50: 0-63, 0 = min, 63 = max)

### KtpSwitchBacklight

- Declaration:  
Visual C:     UINT8 KtpSwitchBacklight(/\*[in]\*/BacklightStat  
  backLight);  
Visual Basic:   KtpSwitchBacklight(ByVal backlight As Integer) As Byte
- Description:  
Turns on/off the backlight of the LC display on the device.
- Arguments:  
BACKLIGHT\_ON, BACKLIGHT\_OFF

**KtpSetScreenSaverTimeOutMin**

- Declaration:  
Visual C:        UINT8 SetScreenSaverTimeOutMin(/\*[in]\*/UINT8  
                  u8\_ScreenSaverTO);  
Visual Basic:    KtpSetScreenSaverTimeOutMin(Byte screenSaverTO  
                  As Byte) As Byte
- Description:  
Sets the timeout value of the screensaver in minutes.
- Arguments:  
0-255, 0 = off, 255 = max

**KtpSetScreenSaverTimeOutSec**

- Declaration:  
Visual C:        UINT8 SetScreenSaverTimeOutMin(/\*[in]\*/UINT16  
                  u16\_ScreenSaverTO);  
Visual Basic:    KtpSetScreenSaverTimeOutSec(Byte screenSaverTO  
                  As Integer) As Byte
- Description:  
Sets the timeout value of the screensaver in seconds.
- Arguments:  
0-65535, 0 = off, 65535 = max

**KtpSetBuzzerVolume**

- Declaration:  
Visual C:        UINT8 KtpSetBuzzerVolume(/\*[in]\*/UINT8 u8\_Volume);  
Visual Basic:    KtpSetBuzzerVolume(Byte volume As Byte) As Byte
- Description:  
Sets the volume of the buzzer.
- Arguments:  
0-16, 0 = off, 16 = max

## Functions for Reading the Configuration

These functions return the current value of the configuration parameters. None of the functions need any parameter. These functions do not enable checking for errors since the return value of the function is the value of the configuration parameter.

### KtpGetBrightness

- Declaration:  
Visual C:       UINT8 KtpGetBrightness(void);  
Visual Basic:   KtpGetBrightness() As Byte
- Description:  
Gets the current brightness value of the LC display.

### KtpGetContrast

- Declaration:  
Visual C:       UINT8 KtpGetContrast(void);  
Visual Basic:   KtpGetContrast() As Byte
- Description:  
Gets the current contrast value of the LC display.

### KtpGetBacklight

- Declaration:  
Visual C:       TKtpBacklightStat KtpGetBacklight(void);  
Visual Basic:   KtpGetBacklight() As Integer
- Description:  
Gets the current status of the background lighting.

### KtpGetScreenSaverTimeoutMin

- Declaration:  
Visual C:       UINT8 GetScreenSaverTimeOutMin(void);  
Visual Basic:   KtpGetScreenSaverTimeOutMin() As Byte
- Description:  
Gets the current timeout value of the screensaver in minutes.



**KtpGetScreenSaverTimeoutSec**

- Declaration:  
Visual C:       UINT8 GetScreenSaverTimeOutSec(void);  
Visual Basic:   KtpGetScreenSaverTimeOutSec() As Integer
- Description:  
Gets the current timeout value of the screensaver in seconds.

**KtpGetBuzzerVolume**

- Declaration:  
Visual C:       UINT8 KtpGetBuzzerVolume(void);  
Visual Basic:   KtpGetBuzzerVoume() As Byte
- Description:  
Gets the current volume value of the buzzer.

## Peripheral Functions

### KtpJoysticksInstalled

- Declaration:  
Visual C:       UINT8 KtpJoysticksInstalled(void);  
Visual Basic:   KtpJoysticksInstalled() As Byte
- Description:  
Returns the number of joystick axes. If no joystick is installed on the device, 0 will be returned.

### KtpWheelsInstalled

- Declaration:  
Visual C:       UINT8 KtpWheelsInstalled(void);  
Visual Basic:   KtpWheelsInstalled() As Byte
- Description:  
Returns 1 if a handwheel is installed on the device, otherwise 0.

### KtpPotilsInstalled

- Declaration:  
Visual C:       UINT8 KtpPotilsInstalled(void);  
Visual Basic:   KtpPotilsInstalled() As Byte
- Description:  
Returns 1 if an override potentiometer is installed on the device, otherwise 0.

### KtpGetJoystickPos

- Declaration:  
Visual C:       UINT8 KtpGetJoystickPos ([out]/TKtpJoystickPos \*p\_jPos);  
Visual Basic:   See function KtpGetJoystickPosEx
- Description:  
Gets the current joystick position.
- Arguments:  
TKtpJoystickPos structure. Each component in the structure may only range between -15 and 15.
- Remarks:  
Calling this function is only allowed if a joystick is installed on the device. If no joystick is installed the values of the components are undefined.

**KtpGetJoystickPosEx**

- Declaration:  
Visual C:        UINT8 KtpGetJoystickPosEx(/[\*out]\*/int \*posX, int \*posY, int \*posZ);  
Visual Basic:    KtpGetJoystickPosEx(ByRef posX As Integer, ByRef posY As Integer, ByRef posZ As Integer) As Byte
- Description:  
Gets the current joystick position.
- Arguments:  
Each component in the structure may only range between -15 and 15.
- Remarks:  
Calling this function is only allowed if a joystick is installed on the device. If no joystick is installed the values of the components are undefined.

**KtpGetJoystickPosRaw**

- Declaration:  
Visual C:        UINT8 KtpGetJoystickPosRaw  
                  (/[\*out]\*/TktJoystickPosRaw \*p\_jPos);  
Visual Basic:    see function: KtpGetJoystickPosRawEx
- Description:  
Gets the actual raw data of the joystick.
- Remarks:  
Calling this function is only allowed if a joystick is installed on the device. If no joystick is installed the values of the components are undefined.

**KtpGetJoystickPosRawEx**

- Declaration:  
Visual C:        UINT8 KtpGetJoystickPosEx(/[\*out]\*/UINT16 \*posX, UINT16 \*posY, UINT16 \*posZ);  
Visual Basic:    KtpGetJoystickPosEx(ByRef posX As Integer, ByRef posY As Integer, ByRef posZ As Integer) As Byte
- Description:  
Gets the actual raw data of the joystick.
- Remarks:  
Calling this function is only allowed if a joystick is installed on the device. If no joystick is installed the values of the components are undefined.

**KtpSetJoystickCalibData**

- Declaration:  
Visual C:       UINT8 KtpSetJoystickCalibData(/\*in\*/TKtpJoystickChannel ch UINT16 rawMin, UINT16 rawCenter ,  
                  UINT16 rawMax, UINT16 calibRange);  
Visual Basic:   KtpSetJoystickCalibData(ByVal joystickChannel As Integer, ByVal rawMin As Integer, ByVal rawCenter As Integer, ByVal rawMax As Integer, ByVal calibRange As Integer);
- Arguments:  
TktJoystickChannel ch: channel to be calibrated.  
UINT16 rawMin: value for smallest raw value  
UINT16 ramCenter: average value for raw data  
UINT16 rawMax: maximum value of raw data  
UINT16 calibRange: maximum range of joystick
- Description:  
Calibrates the axis of the joystick.
- Remarks:  
This function may only be called if a joystick is installed on the device. If no joystick is installed, the value of the components are undefined.

**KtpGetOverridePoti**

- Declaration:  
Visual C:       UINT8 KtpGetOverridePoti(/\*out\*/UINT8 \*p\_pos);  
Visual Basic:   KtpGetOverridePoti(ByRef pos As Byte)
- Description:  
Gets the current value of the override potentiometers.
- Arguments:  
Pointer to the variable containing the current value of the override potentiometer.
- Remarks:  
This function may only be called if a override potentiometer is installed on the device. If no override potentiometer is installed, the value of the components are undefined.

**KtpGetOverridePotiRaw**

- Declaration:  
Visual C:        UINT8 KtpGetOverridePotiRaw(/\*out\*/UINT16 \*p\_pos);  
Visual Basic:    KtpGetOverridePotiRaw(ByRef pos As Byte) As Byte;
- Description:  
Gets the uncalibrated value from the override potentiometer.
- Arguments:  
Pointer to the variable containing the current value of the override potentiometer.
- Remarks:  
This function may only be called if a override potentiometer is installed on the device. If no override potentiometer is installed, the value of the components are undefined.

**KtpGetEnablingDevice**

- Declaration:  
Visual C:        UINT8 KtpGetEnablingDevice(/\*in\*/  
                  TKtpEnablingDeviceCircuit circuit);  
Visual Basic:    KtpGetEnablingDevice(ByRef deviceCircuit As Integer)  
                  As Byte;
- Description:  
Circuit is the value of the enabling switch to be read out when it is pressed completely (panic position).
- Arguments:  
Returns the current value of the enabling switch transferred in the circuit.

**KtpSetPotiCalibData**

- Declaration:  
Visual C:        UINT8 KtpSetPotiCalibData(/\*in\*/ UINT16 rawMin,  
                  UINT16 rawMax, UINT16 calibRange);  
Visual Basic:    KtpSetPotiCalibData(ByVal rawMin As Integer,  
                  ByVal rawMax As Integer, ByVal calibRange As Integer);
- Arguments:  
UINT16 rawMin: value for smallest raw value  
UINT16 rawMax: maximum raw value  
UINT16 calibRange: maximum range of overridepoti
- Description:  
Calibration of override potentiometer.  
This function may only be called if a override potentiometer is installed on the device. If no override potentiometer is installed, the value of the components are undefined.

**KtpGetWheelValue**

- Declaration:  
Visual C:       UINT8 KtpGetWheelValue(/\*out\*/UINT16 \*p\_val);  
Visual Basic:   KtpGetWheelValue(ByRef val As Integer) As Byte
- Description:  
Gets the actual value of the handwheel.
- Arguments:  
Pointer to the variable containing the current value of the handwheel.
- Remarks:  
This function may only be called if a handwheel is installed on the device. If no handwheel is installed, the value of the components are undefined.

**KtpSetWheelValue**

- Declaration:  
Visual C:       UINT8 KtpSetWheelValue(/\*in\*/UINT16 val);  
Visual Basic:   KtpSetWheelValue(ByVal val As Integer) As Byte
- Description:  
Sets the current position of the handwheel to the value transferred in val.
- Arguments:  
val is the value the handwheel should be set to.
- Remarks:  
This function may only be called if a handwheel is installed on the device. If no handwheel is installed, the value of the components are undefined.

## Keypad Functions

For details about the LED numbering (ledNr) needed for the following functions, refer to the chapter „Led Numbering“.

### KtpSetKeyboardLed

- Declaration:  
Visual C:        UINT8 KtpSetKeyboardLed(/\*in\*/UINT8 ledNr,  
                  TLedState ledState);  
Visual Basic:    KtpSetKeyboarLed(Byte ledNr As Byte, Byte ledState  
                  As Integer) As Byte
- Description:  
Switches the set LED to the desired state.
- Arguments:  
UINT8 ledNr: number of LED to be set.  
UINT ledState: state, the LED is set to.

### KtpGetKeyboardLed

- Declaration:  
Visual C:        TLedState KtpGetKeyboardLed(/\*in\*/UINT8 ledNr);  
Visual Basic:    KtpGetKeyboardLed(Byte ledNr As Byte) As Integer
- Description:  
Returns the state of the desired LED.
- Arguments:  
UINT8 ledNr: number of LED.

## Other Functions

These functions perform various actions on the device.

### KtpGetStatus

- Declaration:  
Visual C:        UINT8 KtpGetStatus(/\*[out]\*/Status \*p\_tpuStatus);  
Visual Basic:    KtpGetStatus(ByRef ktpStatus As Byte) As Byte
- Description:  
Gets the start-up state of the handheld terminal.
- Arguments:  
Status \*p\_tpuStatus, pointer to memory area that will receive a copy of the status structure.

### KtpWriteToFlash

- Declaration:  
Visual C:        UINT8 WriteToFlash(void);  
Visual Basic:    KtpGetWriteToFlash() As Byte
- Description:  
Writes the registry of the device to the flash.

### KtpReset

- Declaration:  
Visual C:        UINT8 KtpReset(void);  
Visual Basic:    KtpReset () As Byte
- Description:  
Restarts the device new.

### KtpGetVersionString

- Declaration:  
Visual C:        UINT8 KtpGetVersionString(/\*out\*/TCHAR  
                  \*wszVersionString, UINT16 bufferSize);  
Visual Basic:    KtpGetVersionString (ByRef versionString As String,  
                  ByVal buffSize As Integer) As Byte
- Description:  
Returns the state of the KeTop during start-up.
- Arguments:  
TCHAR \*wszVersionString: pointer to buffer for the versions string  
unsigned int bufferSize: length of transferred buffer.



**KtpGetEEPromData**

- Declaration:  
Visual C:     UINT8 KtpGetEEPromData(/\*out\*/TEEPromData \*data);  
Visual Basic: not implemented.
- Description:  
Read the data from the EEPROM
- Arguments:  
TEEPromData data: data structure for the data contained in the EEPROM

**KtpWriteByteToEEProm**

- Declaration:  
Visual C:     UINT8 KtpWriteByteToEEProm (/\*in\*/UINT16 addr,  
  UINT8 data);  
Visual Basic: KtpWriteByteToEEProm (ByVal addr As Integer,  
  ByVal data As Byte) As Byte;
- Description:  
Writes a byte to the EEPROM location transferred in addr.
- Arguments:  
UINT16 addr: address of memory location  
UINT8 data: data for saving

**KtpReadByteToEEProm**

- Declaration:  
Visual C:     UINT8 KtpReadByteToEEProm (/\*in\*/UINT16 addr,  
  /\*out\*/ UINT8 \*pData);  
Visual Basic: KtpWriteByteToEEProm (ByVal addr As Integer,  
  ByRef pData as Byte) As Byte;
- Description:  
Reads a byte from the EEPROM location transferred in addr.
- Arguments:  
UINT16 addr: address of memory location  
UINT8 \*pData: data from the EEPROM

**KtpLaunchTouchScreenCalibApp**

- Declaration:  
Visual C:     UINT8 KtpLaunchTouchScreenCalibApp (void);  
Visual Basic: KtpLaunchTouchScreenCalibApp () As Byte;
- Description:  
Starts the touch-screen calibration tool.

**KtpGetTemperature**

- Declaration:  
Visual C:     UINT8 KtpGetTemperature (void);  
Visual Basic: KtpGetTemperature () As Integer;

- Description:  
Returns the current temperature of the KeTop.

### KtpPlaySound

- Declaration:  
Visual C:        UINT8 KtpPlaySound (/\*in\*/ UINT16 soundNr);  
Visual Basic:    KtpPlaySound (ByVal soundNr As Integer) As Byte;
- Description:  
Plays the system sound in soundNr.
- Arguments:  
UINT16 soundNr: number of system sound.

### KtpDoBeep

- Declaration:  
Visual C:        UINT8 KtpDoBeep (/\*in\*/ UINT8 beepTime10ms);  
Visual Basic:    KtpDoBeep (ByVal beepTime10ms As Byte) As Byte;
- Description:  
Is active for beepTime10ms \* 10 ms
- Arguments:  
UINT8 beepTime10ms: duration of beeps in n\* 10ms

### KtpGetVariantData

- Declaration:  
Visual C:        UINT8 KtpGetVariantData (/\*out\*/  
                  TKtpVariantData \*data);  
Visual Basic:    function not implemented
- Description:  
Reads the device configuration from the EEPROM.
- Arguments:  
TKtpVariantData data: data structure for VariantData.

## Functions for Subscribing Events

These functions are used to subscribe/unsubscribe callback functions for different events. Joystick, override potentiometer, handwheel and keypad can be used as events.

### KtpInstallWheelEventCallback

- Declaration:  
Visual C:        UINT8 KtpInstallWheelEventCallback  
                  (/in\*/TktpWheelEventCallback pWheelProc, int \*cookie);  
Visual Basic:    function not implemented
- Description:  
Subscribes a callback function for the WheelEvent and returns an index (cookie) for the callback function.
- Arguments:  
TKtpWheelEventCallback pWheelProc: callback function to be called when the event occurs.  
int cookie: The index for the callback function is required for removing the callback function.

### KtpRemoveWheelEventCallback

- Declaration:  
Visual C:        UINT8 KtpRemoveWheelEventCallback (int cookie);  
Visual Basic:    function not implemented
- Description:  
Removes the WheelEventCallback function.
- Arguments:  
int cookie: index for the callback function to be removed.

### KtpInstallOvrEventCallback

- Declaration:  
Visual C:        UINT8 KtpInstallOvrEventCallback  
                  (/in\*/ TKtpOvrEventCallback pOvrProc, /out\*/ int \*cookie);  
Visual Basic:    function not implemented
- Description:  
Subscribes a callback function for the OverrideEvent and returns an index (cookie) for the callback function.
- Arguments:  
TKtpOvrEventCallback pOvrProc: callback function to be called when the event occurs.  
int cookie: The index for the callback function is required for removing the callback function.

**KtpRemoveOvrEventCallback**

- Declaration:  
Visual C:       UINT8 KtpRemoveOvrEventCallback (int cookie);  
Visual Basic:   function not implemented
- Description:  
Removes the OverrideEventCallback function.
- Arguments:  
int cookie: index for the callback function to be removed.

**KtpInstallKbdEventCallback**

- Declaration:  
Visual C:       UINT8 KtpInstallKbdEventCallback  
                  (/\*in\*/ TktpKbdEventCallback pKbdProc, /\*out\*/  
                  int \*cookie);  
Visual Basic:   function not implemented
- Description:  
Subscribes a callback function for the KeyboardEvent and returns an index (cookie) for the callback function.
- Arguments:  
TktpKbdEventCallback pKbdProc: callback function to be called when the event occurs.  
int cookie: The index for the callback function is required for removing the callback function.

**KtpRemoveKbdEventCallback**

- Declaration:  
Visual C:       UINT8 KtpRemoveWheelEventCallback (int cookie);  
Visual Basic:   function not implemented
- Description:  
Removes the WheelEventCallback function.
- Arguments:  
int cookie: index for the callback function to be removed

**KtpInstallJoyEventCallback**

- Declaration:  
Visual C:       UINT8 KtpInstallJoyEventCallback (/\*in\*/  
                  TktJoyEventCallback pJoyProc, /\*out\*/ int \*cookie);  
Visual Basic:   function not implemented
- Description:  
Subscribes a callback function for the JoystickEvent and returns an index (cookie) for the callback function, if a joystick is installed.
- Arguments:  
TktJoyEventCallback pJoyProc: callback function to be called when the event occurs.  
int cookie The index for the callback function is required for removing the callback function.

**KtpRemoveJoyEventCallback**

- Declaration:  
Visual C:       UINT8 KtpRemoveJoyEventCallback (int cookie);  
Visual Basic:   function not implemented
- Description:  
Removes the JoyEventCallback function, if a joystick is installed.
- Arguments:  
int cookie: index for the callback function.

## Program for Starting the Application and KeTop API

This chapter describes the program for starting the application and the KeTop API. The program is contained as a StartAPI.exe in the Windows directory.

### Functional description

The StartAPI program is started by an entry in the Startup directory or by an entry in the registry under [HKEY\_LOCAL\_MACHINE\init] when the KeTop is started.

When the program is started, the KeTop API will be initialized first and then the programs entered under [HKEY\_LOCAL\_MACHINE\Autostart] will be started.

The program remains invisible in the memory and handles different input devices, e.g. override potentiometer and handwheel.

### Registry entries

[HKEY\_LOCAL\_MACHINE\Autostart] contains the entries that are read out by the program.

The following entries are possible:

- **Startx**  
string identifying the program to be started. x is a number between 1 and 255.
- **Paramsx**  
string containing the parameters for the program identified in **Startx**. x must have the same value as in **Startx**.
- **Delayx**  
DWORD containing the waiting time in milliseconds until the next program is started. x must have the same value as in **Startx**.
- **DependStartx**  
array of 10 bytes containing the numbers of the programs which must have been started so that the program x may start.  
If there is no such entry or all bytes are 0 then the program x starts and no other programs must have been started.
- **DependEndx**  
array of 10 bytes containing the numbers of the programs which must have been stopped before the program x starts.  
If there is no such entry or all bytes are 0 then the program x starts without waiting of any other programs.

## Example

Example of a possible configuration in the registry:

```
[HKEY_LOCAL_MACHINE \ Autostart]
Start1      = "cmd.exe"
Params1     = "/c copy \ipsm\windows\*. * \windows"
Delay1      = dword:0x000003e8 (1000)

Start10     = "pvbload.exe"
Params10    = "\ipsm\vb\TestProjectVB1.vb"
Delay10     = dword:0x00002710 (10000)

Start20     = "SetTime.exe"
Params20    = ""
Delay20     = dword:0

Start30     = "ReadCorrectTime.exe"
Params30    = ""
Delay30     = dword:0
DependStart30 = hex: a, 0, 0, 0, 0, 0, 0, 0, 0, 0
DependEnd30  = hex:14, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

When the StartAPI is started, the KeTop API will be initialized first and then the program cmd.exe will be called with the parameters "/c copy \ipsm\windows\\*. \* \windows".

```
cmd.exe /c copy \ipsm\windows\*. * \windows
```

This program copies the data from the directory \ipsm\windows to the normal Windows directory.

After the call of cmd.exe, the system waits for 1000 milliseconds (Delay1) until the next program is started even if the cmd.exe has not been completed before.

After the delay, the program pvbload.exe will be started with the parameters "\ipsm\TestProjectVB1.vb".

```
pvbload.exe \ipsm\TestProjectVB1.vb
```

This program starts the VisualBasicScript TestProjectVB1.vb.

After a waiting time of 10000 milliseconds (Delay10), the program with the next higher entry will be started.

SetTime starts a program for setting the time. This program will be started without parameter and delay.

Afterwards it will be checked if the program 10 (DependStart30 a,.. ) has been started and the SetTime program (DependEnd30 14,..) has been fin-

ished already. If all conditions are met the program ReadCorrectTime will be started.

If no more entries are available, the starting of the applications will be finished.

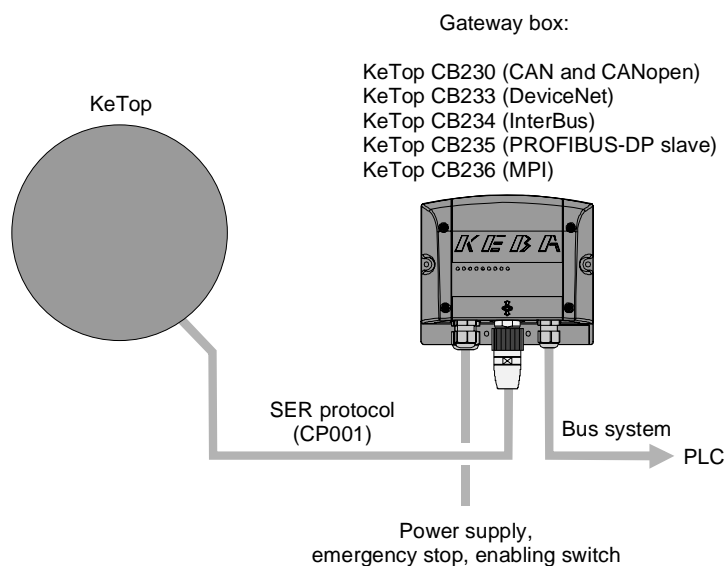
Programs started by the StartAPI will not be finished.



## SER protocol

### General

On the one hand, the SER protocol is used for the secure packet transport at the character level between the KeTop and Gateway box and, on the other hand, it supports the secure net data exchange between the two user programs on the KeTop and on the PLC (in the case of the T100 and T50 or with the CP001 data profile) with the handshake mechanism.



SER protocol between KeTop and gatewaybox

## SER Protocol Details

This chapter is intended for SW development engineers who are developing a handling software on a PLC to enable a direct connection (without Gateway box) from a KeTop to a PLC.

The SER-protocol is a simple serial protocol operated in the full duplex mode. The user data are transported as follows:

STX | n data bytes | ETX | BCC →                      ←      ACK / NAK

The data length n can be defined in the configuration phase, but remains constant during operation.

The checksum (BCC) is built by an XOR logic of all message characters beginning with STX and ending with ETX.

After sending the STX character, the remaining part of the message must be transmitted within a certain time. The monitoring time on the receive side is calculated from the number of characters multiplied by 2ms. That means, in case of a set data length of  $n = 8$  bytes, the monitoring time will be  $(8 + 3) \times 2 = 22\text{ms}$ .

After sending the BCC character, the send station waits for a maximum of 50 ms for the acknowledgement character ACK (positive confirmation) or NAK (negative confirmation) from the partner station.

If no acknowledgement character is received within this time, a new message can be sent.

The receive station will only check if the message is correct after having received a valid STX character. Each other character it receives before will be ignored no matter whether the character is valid or not. The only exception is the NAK character. After receipt of this character, the current send message will be repeated (see below).

That means only if a valid STX character has been recognized, a message error will be acknowledged with NAK. The following error sources are possible:

- The monitoring time has expired before the message was received completely.
- The message format is not correct ([n+2]nd character is no ETX)
- The BCC is not correct.

Once a station receives a NAK character, it will repeat the message sent last. If an incorrect acknowledgement character or no acknowledgement character is received, the message will not be repeated.

A message already sent can be requested again later by sending a NAK character (even if it has already been acknowledged with ACK). The number of these NAK repetitions is NOT limited.

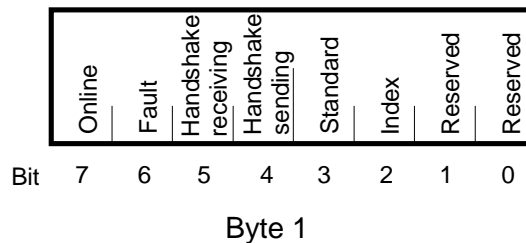
The gateway utilizes this mechanism for the life monitoring of the KeTop device. If the gateway has not received any message nor any acknowledgement from the KeTop terminal for a time to be set (e.g. 500ms), it will send a NAK character to which the KeTop responds with a send message. This may be the message sent last (even if it has already been acknowledged with ACK) or a new message if the message sent last has already been acknowledged with ACK before (within 50 ms). If the gateway does not receive any valid character as a response to this "life monitoring NAK" within 50 ms, the connection to the KeTop device is probably interrupted. In this case, the gateway must inform the PLC accordingly by resetting bit 7 (online bit) in the first data byte (handshake information) and then sending the data to PLC. This is the only (!) situation in which the gateway modifies the transferred data.

**Notice:**

The 50ms response time to the NAK character is only controlled after the special "life monitoring NAK" has been sent. In case of "protocol error NAKs" (wrong BCC, etc.) this time will not be monitored. That means, as long as (valid) characters are exchanged between the two stations (no matter whether protocol errors occur in the meantime), resetting online bits is not allowed. Only if no valid character has been received from the KeTop, this "life monitoring NAK" will be sent. If, after that, no valid character (of any type) is received for 50 ms, the online bit will be toggled thus signaling to the PLC that the connection to the KeTop is interrupted.

## Handshaking

Because of the different data transmission rate between serial side and bus side as well as the different data transfer mechanisms on the different bus systems, a controlled consistent data exchange between both end stations must be ensured. This is achieved by a corresponding **handshake information** which is contained **in the first byte** of the transmitted data.

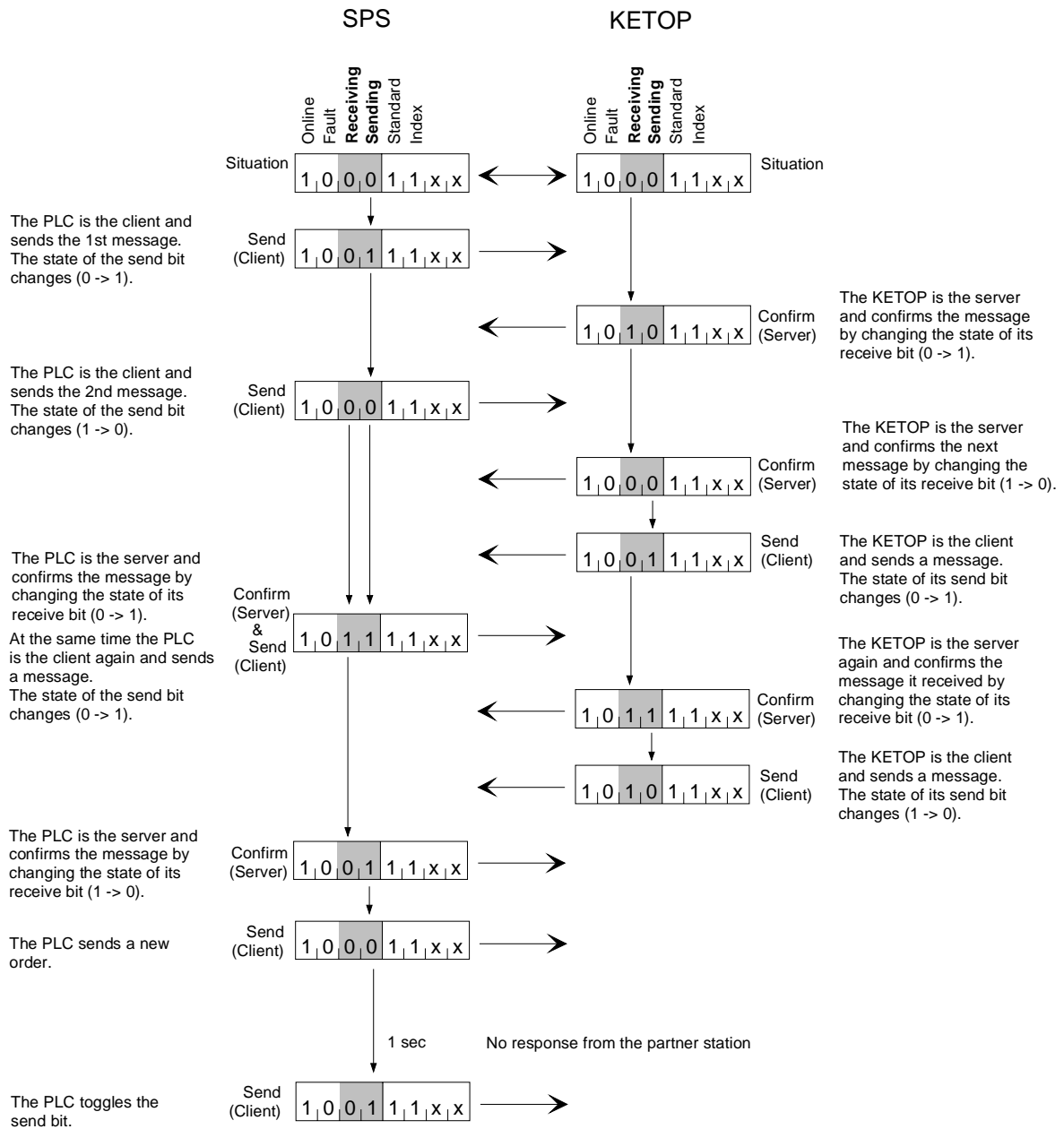


Structure of handshake byte

### Description of bits:

- Online (bit 7)** This bit indicates that the device is ready for receiving messages. If the device is not online, no messages may be sent. Irrespective of that, the device which is not online may send data. The online bit and the fault bit do not depend on each other.
- Fault (bit 6)** This bit indicates a fault on the device. If the device is online a fault code can be requested.
- Handshake receive (bit 5)** This bit confirms the receipt of a message. The state of this bit changes each time a message is received:  
The state of the bit changes from 0 to 1 when the first message is received, and from 1 to 0 when the next message is received.
- Handshake send (bit 4)** This bit signalizes the sending of a new message. The state of this bit changes each time a message is sent:  
The state of the bit changes from 0 to 1 when the first message is sent, and from 1 to 0 when the next message is sent.
- Standard (bit 3)** This bit defines the assignment of the data field for the process data transmission. In this case, the significance of the PD index is standardized. If this bit is not set, the significance of this field is manufacturer-specific. (This byte is not evaluated in the case of the CP001 data profile!)
- Index (bit 2)** This bit defines the existence of an index field in byte 2 of the data field of the indirect process data channel. If the standard bit is set the index will be transmitted in byte 2 in the index field. If the standard bit is not set the index is manufacturer-specific. (This byte is not evaluated in the case of the CP001 data profile!)

The following picture shows an example for the handshaking between two end stations (KeTop and PLC).

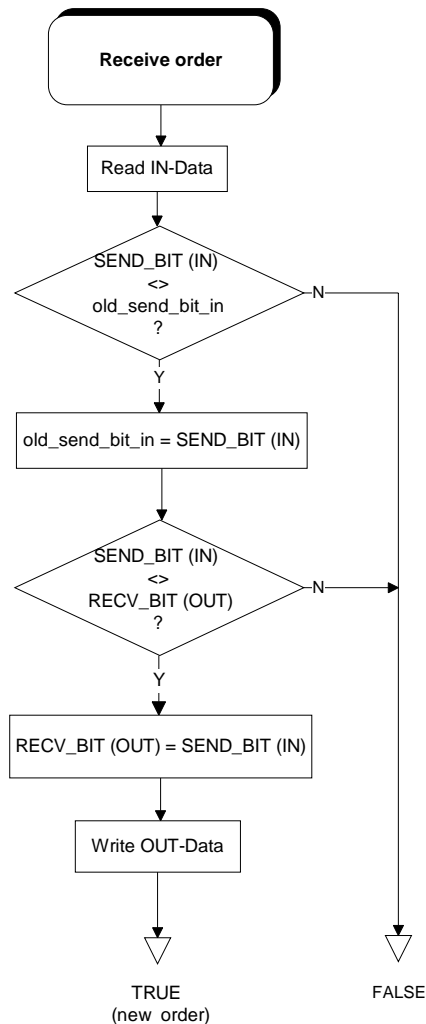


As shown in this example, the confirmation and the sending of a new message can also be performed via a transaction. That means, the send and the receive bit are toggled simultaneously.

If the partner station does not respond within 1 second to a message received by toggling its receive bit, the send station will toggle its send bit to its previous status again.

### Suggestions of implementation

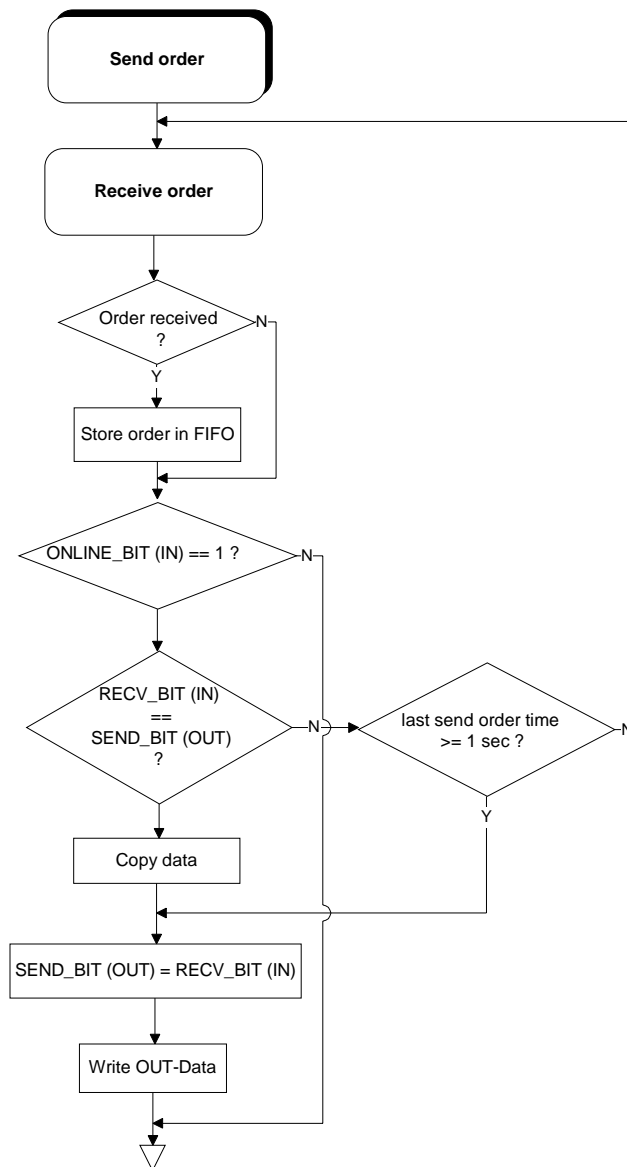
Receive order



Receive order

## Send order

Before an order is sent, it should be checked if there is already a pending order of the partner station. If yes, it should be given priority. Following that, the send station checks if the receive bit of the partner station has the same state as its own send bit. If this is the case the device may send data. The data to be sent will be loaded into the output buffer. Following that, the send bit of the send station will be inverted to indicate the sending. The handshake of the partner station is monitored. If the counterpart station does not respond within 1 s, the send bit will be toggled to its previous state (see Functionality of server).

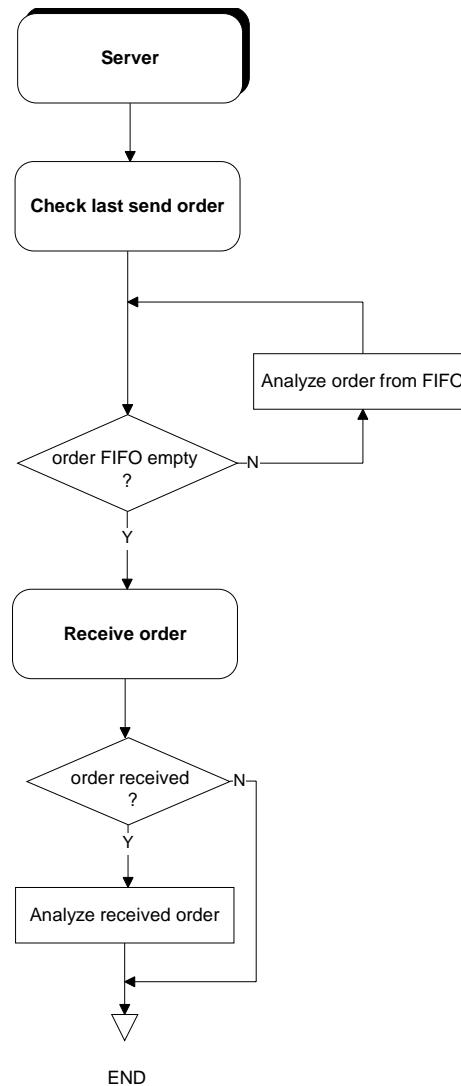


Send order

Server:

This function must be called cyclically (e.g. within a task). The following steps are performed:

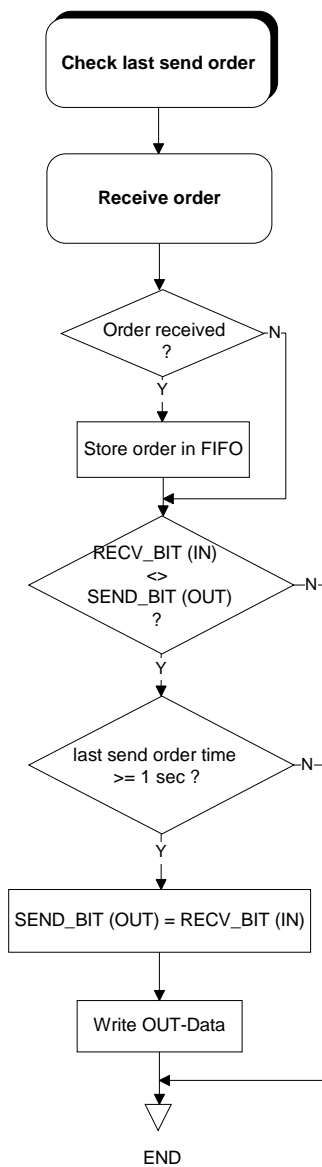
- Checking of acknowledgement for last send message (if necessary, the send bit will be toggled)
- Execution of temporarily stored orders
- Execution of pending order from the partner station



Server



Checking of acknowledgement for last send message (if necessary, the send bit will be toggled)



Checking of acknowledgement

## CP001 Data Profile

The CP001 profile is defined by KEBA. The length of the CP001 messages depends, among others, on the data length available on the bus system. Once set in the configuration phase, the message length (l) remains constant.

*Structure of CP001 data messages:*

Length in bytes	Contents	
1	Handshake	Header
1	Index (command)	
2	Address	
1	Number of data bytes (n)	
1	Number of I/O bytes (m)	
m	I/O bytes	Data
n	Data bytes	
l-m-n-6	0	

$(m + n + 6) \leq l$  must apply.

The 16-bit value for the address is transmitted in the Motorola format (high-byte before low-byte).

The I/O bytes are directly copied into the PLC process data memory (e.g. KeTop keymap). The I/O bytes can be evaluated from each message received – independently of the index – because the sender must ensure to copy the current process data into this range.

The data bytes are to be evaluated according to the index (or command).

Command values (index byte):

	Request	Response
NOP (I/O data only)	0x00	-
Read	0x01	0x41
Write	0x02	-
Write – AND	0x03	-
Write – OR	0x04	-
Write – XOR	0x05	-
Error code	0x06	-

There is no difference between handling request messages and handling response messages.

At error codes messages, the error code with the length of one byte is contained in the data byte part.

Significance of error codes:

0	Not used because means "without error"
01H .. 7FH	To be assigned by the developer of the PLC driver
80H .. FFH	Reserved for extension

## CP001 software interface

### Functions

All functions return a Boolean value that specifies whether the function can be carried out free of error. If an error occurs, the error can be requested with the function SerCp001GetLastError.

#### Opening the serial channel

Opens a port and enables access for the extended access functions.

<b>Function</b>	bool SerCp001Open(const char *pPortName, unsigned long baudRate)
<b>Parameter</b>	pPortname Channel to be opened baudRate Baud rate
<b>Return value</b>	true OK false Error
<b>Addition</b>	All other parameters of the interface are specified by the CP001 protocol and cannot be parameterized.

#### Closing the serial channel

Closes the serial port again and further accesses are no longer possible.

<b>Function</b>	bool SerCp001Close(void)
<b>Parameter</b>	-
<b>Return value</b>	true OK false Error

#### Configuring the serial channel

Configures the maximum permitted length of a CP001 frame.

<b>Function</b>	bool SerCp001Config(unsigned long frameLenght)
<b>Parameter</b>	frameLenght Maximum length of a CP001 packet (incl. CP001 header(6Bytes), I/O, data)
<b>Return value</b>	true OK false Error
<b>Addition</b>	The maximum packet length must be divisible by 16 or 8.

#### Reading a CP001 packet

Reads in a CP001 frame and provides data and I/O.

<b>Function</b>	bool SerCp001Read(UINT8 *cmd, UINT16 *dataAddr, UINT8 *data, UINT8 *dataLength, UINT8 *io, UINT8 *ioLength, unsigned long timeOutMs)	
<b>Parameter</b>	cmd	Index or command
	dataAddr	Start address of the user data in memory
	data	Data array with data bytes
	dataLength	Length of the user data
	io	Data array with I/O bytes
	ioLength	Length of I/O data
	timeOutMs	Timeout for reading
<b>Return value</b>	true OK false Error	
<b>Addition</b>	The maximum packet length must be divisible by 16 or 8.	

### Writing a CP001 packet

The data and I/O handed over are written to the port.

<b>Function</b>	bool SerCp001Write(UINT8 cmd, UINT16 dataAddr, UINT8 *data, UINT8 dataLength, UINT8 *io, UINT8 ioLength)	
<b>Parameter</b>	cmd	Index or command
	dataAddr	Start address of the user data in memory
	data	Data array with data bytes
	dataLength	Length of user data
	io	Data array with I/O bytes
	ioLength	Length of I/O data
<b>Return value</b>	true OK false Error	

### Interrogating a version/revision

The variables handed over are filled with the version string, and version/revision numbers.

<b>Function</b>	bool SerCp001Version(char *versionName, UINT8 *version, UINT8 *revision)	
<b>Parameter</b>	versionName	Version string
	version	Version number
	revision	Revision number
<b>Return value</b>	true OK false Error	
<b>Addition</b>	The version is incremented if any incompatibility with the previous version occurs. All other program changes only cause the revision to be incremented.	

**Installing the callback function**

In order to respond to a received frame, a callback function must be installed that is called up in the case of an incoming frame.

<b>Function</b>	<code>bool SerCp001RecvCallback(TRecvEventProc eventProc)</code>
<b>Parameter</b>	<code>eventProc</code> Pointer auf the callback function
<b>Return value</b>	<code>true</code> ok <code>false</code> error
<b>Pointer type</b>	<code>void (*TRecvEventProc)(void)</code>
<b>Addition</b>	If a message is received from the host, the Callback function is called by the driver. It is then possible to read the received data with "Reading a CP001 packet."

**Interrogating an error code**

If an error of the functions of the CP001 driver occurs, the precise error can be interrogated with this function. For details see "Error codes".

<b>Function</b>	<code>DWORD SerCp001GetLastError)(void)</code>
<b>Parameter</b>	-
<b>Return value</b>	Error number from CP001 driver

## Error codes

### General error

<b>CP001_NO_ERROR</b> (0x20000000)	No error occurred.
<b>CP001_PORT_ALREADY_OPEN</b> (0x20000001)	Port cannot be opened because it is already open.
<b>CP001_PORT_NOT_OPEN</b> (0x20000002)	No access to the port because this is not yet open.
<b>CP001_FRAMELEN_TO_LONG</b> (0x20000003)	The data to be transferred are longer than the maximum configurable length.
<b>CP001_INVALID_FRAMELEN</b> (0x20000004)	Frame length must be divisible by 16 or 8.
<b>CP001_HOST_NOT_ONLINE</b> (0x20000005)	An attempt was made to send to a host that is available but is not online.
<b>CP001_CALLBACK_ALREADY_EXISTS</b> (0x2000000E)	The Callback function has already been installed.

### Transmission protocol

<b>CP001_SER_CRC_FAILS</b> (0x20000006)	When receiving a CP001 frame, a CRC error was detected.
<b>CP001_SER_TIMEOUT</b> (0x20000007)	A timeout during reception was detected (ACK/NAK was sent too late).
<b>CP001_SER_UNKNOWN_CMD</b> (0x20000008)	An unknown CP001 command.
<b>CP001_SER_NO_RESPONSE</b> (0x20000009)	An attempt was made to send to a non-existent host.
<b>CP001_SER_HOST_UNSYNC</b> (0x2000000A)	Protocol between host and Ketop no longer synchronized.
<b>CP001_SER_TOO_MANY_CONFLICTS</b> (0x2000000F)	Occurs if host and Ketop attempted to re-establish the connection at the same time (> 10).

### Server error

<b>CP001_SERVER_RECV_FIFO_EMPTY</b> (0x2000000B)	An attempt was made to read when no data are in the buffer.
<b>CP001_SERVER_RECV_FIFO_FULL</b> (0x2000000C)	The internal FiFo is full, frames can no longer be saved.
<b>CP001_SERVER_THREAD_CREATING</b> (0x2000000D)	Internal error during creation of a thread

## Registry entries

All entries from the CP001 driver in the registry are stored in the directory "Software\\Driver\\CP001-DLL". When the driver is started, these entries are created with the default value if this entry is not yet located in the registry.

## Tracing

Describes the word which specifies what should also be written. The output is made to a separate trace window. Please note that this output changes the time response of the driver. This is used primarily for troubleshooting. The different level values can also be used in combination.

<b>Entry</b>	"TraceWord"	
<b>Default</b>	0x0000	
<b>Level values</b>	0x00000001	Frame data are output
	0x00000002	Special characters (ACK/NACK/STX...) are output
	0x00000004	Serial settings are output
	0x00000008	CP001 protocol outputs
	0x00000010	Server outputs
	0x00000020	Diverse test outputs

## FiFo capacity

Specifies how many frames can be contained at the same time in the receiving FiFo.

<b>Entry</b>	"ServerFIFODepth"
<b>Default</b>	64

## Thread priority

Sets the internal priority of the running thread of the CP001 driver.

**IMPORTANT:** This entry should only be changed if it is known precisely how thread priorities in WindowsCE are to be used! It should only be used for tests!

<b>Entry</b>	"ThreadPriority"
<b>Default</b>	251



## Test tool (VC++ Demo)

### Requirements

The reader must have knowledge of object-oriented programming, embedded C++ and MFC. The graphs are displayed in UML standard according to OMT.

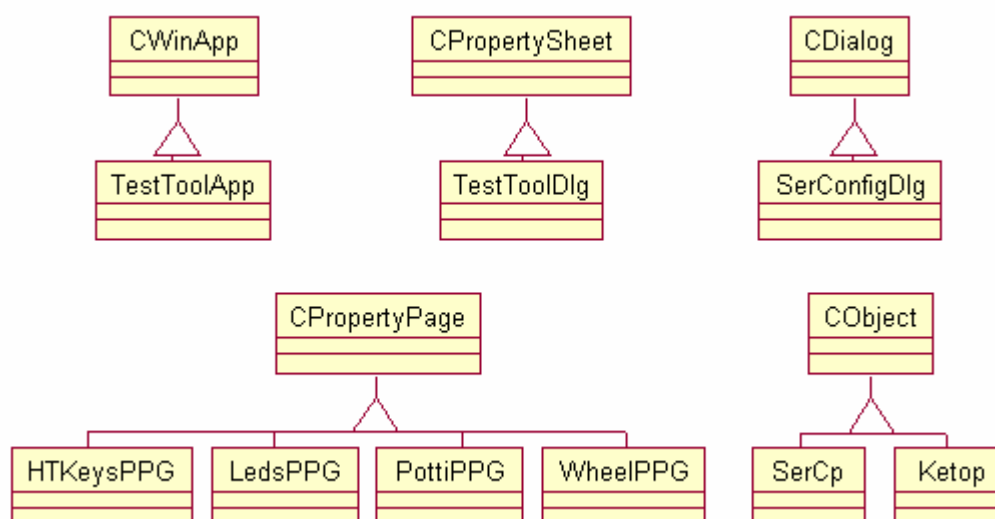
### Task of the test tool

The test tool is a C++ demo and is used to display values that are made available by the KetopAPI, such as wheel values, potentiometer values, key states and LED states, on the screen. These values are transmitted to the control by the SER protocol (CP001 data profile), immediately read out again from the control and also displayed on the screen.

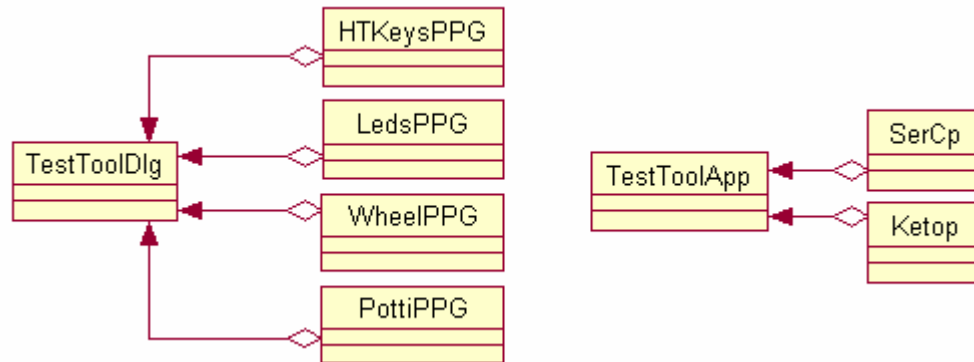
This test tool has been realized in embedded Visual C++ 3.0® with the application of the MFC® (for WinCe).

### Description of the classes

#### Class hierarchies



## Class relations

**TestToolApp**

This class is the actual application that is displayed on the screen. The Init method creates and administrates the dialog (represented by the class TestToolDlg). So that the application can be accessed in a simple way during the running time of the program, this class always contains a member variable at both the SerCp and Ketop levels of abstraction.

**TestToolDlg**

This dialog class contains the individual tabs that are realized by the PropertyPage classes.

**HTKeysPPG**

This class represents the HTKeys tab and is used for visualization of the keystrokes.

**LedsPPG**

This class is responsible for administration of the LEDs and is displayed as the LED tab.

**PottiPPG**

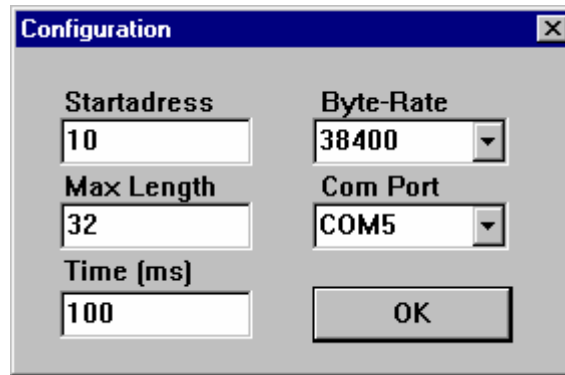
This class takes care of all values of the potentiometer and is visualized in the Potentiometer tab.

**WheelPPG**

This class displays all changes to the wheel in the Wheel tab.

**SerConfigDlg**

This class implements the functionality of the "configuration dialog" (see following figure) that is displayed on the screen when the application is started. This enables the user to enter his own port configuration.



Since the values of this dialog must already be available for the initialization of the port, the dialog is held by the SerCp class as, from a design point of view, it fits best here.

### SerCp

This class forms a level of abstraction above the CP001 driver and should simplify communication with the control for the application.

The Constructor loads the SerCp001.dll. Moreover, function pointers to the method of the SerCp001.dll that is used are immediately created and the Callback function that is used for reading out values to the control, is hung up. The Destructor closes the port and releases the Dll. This class provides a "simplified" interface for communicating with the control.

### Ketop

This class also forms a level of abstraction above the KetopAPI and, thus, should simplify access to it. It provides a somewhat "simplified" interface to the outside world.

The Constructor loads the KetopAPI.dll. Also in this case, function pointers to the required method are retrieved (however, in a somewhat different manner than in the SerCp class) and a Callback function, which is called automatically by the KetopAPI during modification, is hung up.

## Procedure using the example of the test tool

The Init of the TestToolApp class firstly initializes the member variable (mSerCp) of the SerCp class that sees to the configuration of the driver. At the same time, the configuration dialog (SerConfigDlg) is called, the entered values are stored and, with them, the port to the control is opened and configured. The TestToolDialog is then called (in the TestToolApp class) and displayed on the screen. This starts with the start of a timer that issues a cyclical write command with the read command to the control. (Without the timer, the application would never be informed about modifications to the control because, in this test example, the control never becomes active by itself.)

If, for example, the wheel is now rotated, the Callback function of the Keto-pAPI is called automatically. There, a Windows message is sent, and received and processed by the test tool application which, in this case, is synonymous with a graphical update of the tab that is currently active. This modified value is then sent to the control by a write command.

The started timer cyclically informs the control that it would like to read. The control calls the Callback function registered with it and thus triggers a Window message. This, in turn, is received by the application. In the next step, a read command is sent to the control and the tab that is currently active is determined. The modified values are handed over to this tab and only this tab carries out an update of the screen.

### **General procedure for communicating with the control using the SerCp001.dll**

1. Load the library (SerCp001.dll)
2. Retrieve the function pointers to the required method
3. Install the Callback function
4. Open and configure the port
5. Use the methods that are made available by the DLL
6. Close the port again
7. Release the library

## zenOn visualization software

The zenOn visualization software from the COPA-DATA company and available from KEBA has two different variants:

KeTop PT001:	Visualization software with software license
KeTop PT002:	Visualization software with hardware dongle

### Notice


- Basically, the zenOn visualization software is described in the COPA-DATA user manuals (these are also installed during the zenOn installation). The following chapters only describe specifics of the zenOn visualization software in co-operation with a KeTop.

## Installation

For the successful installation of the zenOn visualization software, **three steps** are necessary. To install the software, use the zenOn CD from the company, COPA-DATA:

- ▶ Installation of the zenOn editor by calling up the **setup.EXE** file.
- ▶ Installation of the driver required for Windows CE by calling up the **zenOn\_CE\_Setup.EXE** file (**absolutely necessary!**).
- ▶ To operate the zenOn editor in the respective full version, this must be licensed:

<b>Licensing for KeTop PT001:</b> To activate the software license, it is necessary to request a valid activation number from the COPA-DATA company and to enter this together with the serial number:	<b>Licensing for KeTop PT002:</b> To activate the hardware license, enter the activation and serial number enclosed with the dongle:
▶ File -> General configuration -> "Serial number..."	



## zenOn driver available for Windows-CE

This table gives an overview of each zenOn driver that is available for Windows-CE (see also chapter "Driver selection").

Driver	Control	Connection type			Remark
		Serial	TCP/IP	Other	
3964r32.dll	Various controls Siemens S5, Sauter, etc.	x			3964R protocol
3S_Arti.dll	Codesys Soft PLC - TCP/IP or direct communication local		x		
AllenDF1.dll	Allen Bradley Allen DF1 protocol	x			
BeckhTc32.dll	Beckhoff TwinCAT with TCAds interface		x		
CIFMPI32.dll	Hilscher CIF cards on Siemens MPI or Netlink adapter MPI via TCP/IP		x		NetLink required!
CP001_32.dll	KEBA specific protocol (CP001)	x			
dataTaker32.dll	Data logger from Datataker	x			
Euchner32.dll	Euchner Ident system - access control	x			
HITACHI.dll	Hitachi controls	x			
HMI32.dll	Siemens MPI with HMI adapter	x			HMI adapter required!
KETOP.dll	KEBA, specific KETOP functions (hand wheel, potentiometer, ...)				No communication driver!
KOYO32.dll	Koyo PLC under CE only serial, also TCP/IP on the PC	x			
logix32.dll	Allen Bradley Contrologix 5550		x		
MELSECA.dll	Mitsubishi Melsec A protocol	x			
MINI32.dll	B&R Mininet			Mininet	
Mitsubishi_FX32.dll	Mitsubishi FX controls	x			
MODICON32.dll	Schneider Unitelway protocol	x			
MODRTU32.dll	Modbus RTU. Supported by a large number of many controls. Serial and TCP/IP	x	x		
OMRON32.dll	Omron PLC	x			
OMR_FINS.dll	Omron Fins protocol	x			
Opc2Cli32.dll	OPC Client. Only local OPC Server. No DCOM				
PCS807_32.dll	Lauer PCS 807 box for Profibus DP	x			Lauer box required!
PCS812.dll	Lauer PCS 812 box for MPI	x			Lauer box required!
PSS32.dll	Pilz safety control	x			
PSUNI32.dll	Klöckner Möller	x			
S432.dll	Schiele 400er series	x			
S5PG32.dll	Siemens S5 PG interface	x			Attention: 20mA-CL!
S7TCP32.dll	Siemens S7 via TCP/IP		x		
S800_32.dll	Schiele 800er series	x			
SAIA_Directly32.dll	Saia Serial, MBus and TCP/IP	x	x	MBus	
SAPP32.dll	Synthesi control	x			
SIMUL32.dll	Simulator				No communication driver!

**Remark:**

In each case, different serial interfaces (RS-232-C, RS-422-A, RS-485, 20mA-CL) are available on the various controls.

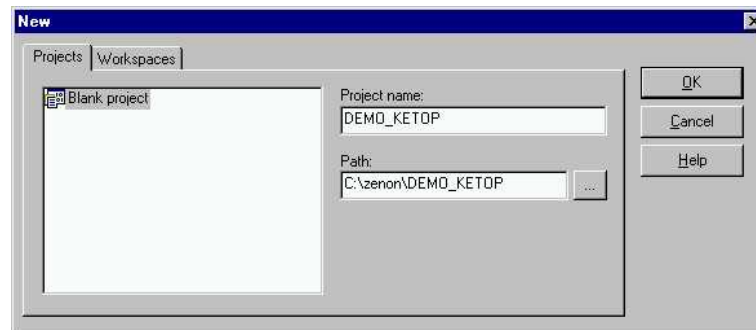
Consequently, it is always necessary to check whether the KEBA handheld terminal can be connected directly!

## Guide to the creation of a new project

This chapter briefly describes the most important steps required to create a new project. To do so, the KeTop SK001 Startkit CD includes an example project in the directory\DEMOS\zenOn\.

### Creating a new project

- ▶ File -> New:  
Input of the project name, e.g.: DEMO\_KETOP

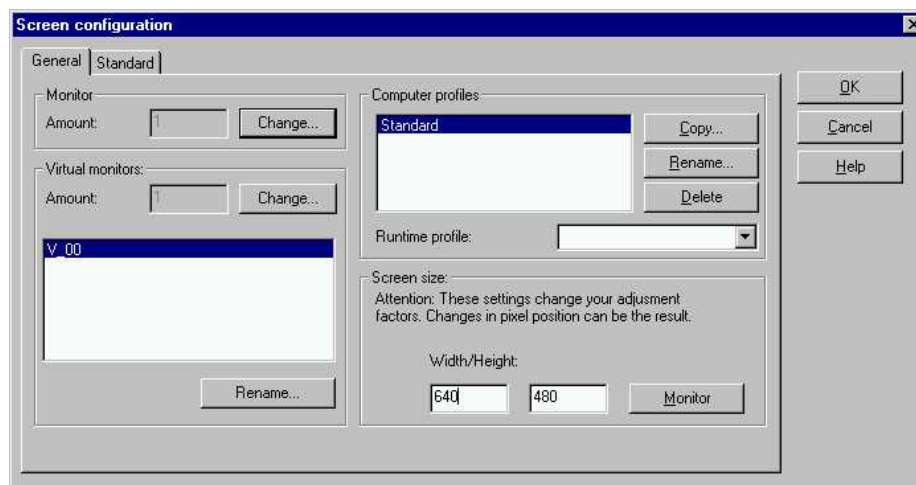


This also creates an active work area at the same time.

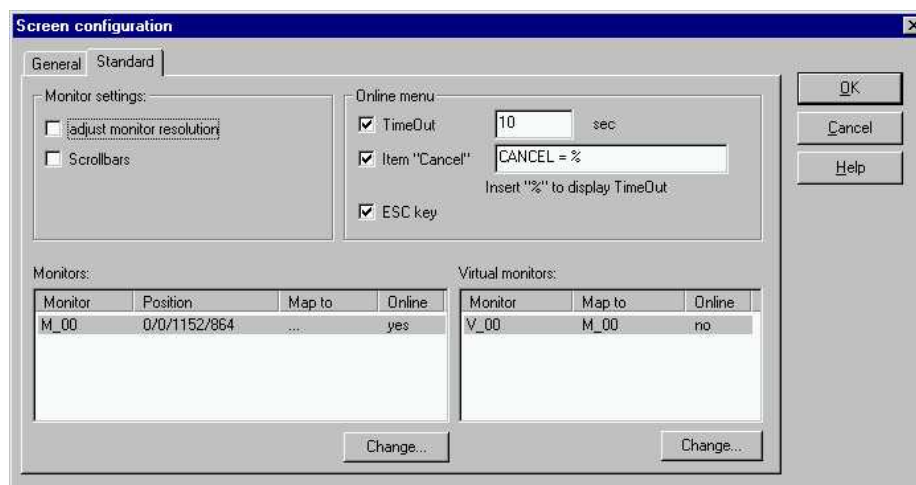


## Configuration of the display size

- ▶ File -> Project configuration -> Monitor administration -> Display configuration
- ▶ Enter the effective size of the KeTop display (e.g. 640 x 480 for VGA) in the "Screen size" field:



- ▶ Change to the "Standard" tab and deactivate the checkbox for "Adjust monitor resolution":

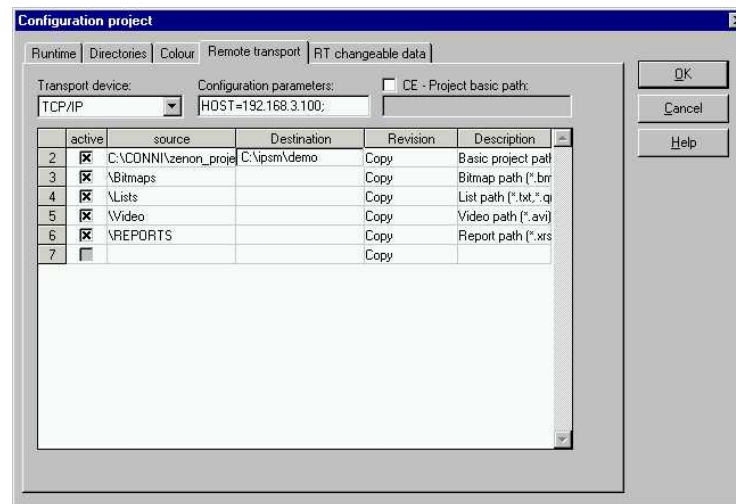


## Configuration of the project

- ▶ File -> Project configuration -> Project -> Configuration Project  
The following settings must be made here:

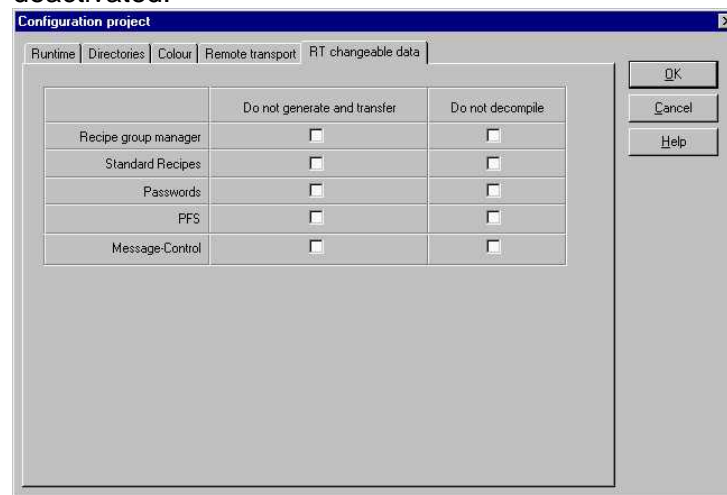
- ▶ "Remote Transport" tab:  
"Transport device" input field:  
Select the interface (serial or Ethernet)

"Configuration parameter" input field:  
Setting up the respective parameters for the interface  
(COM port of the programming PC or IP address of the KeTop).



In addition, the target directory of the project on the KeTop must also be entered in the table.

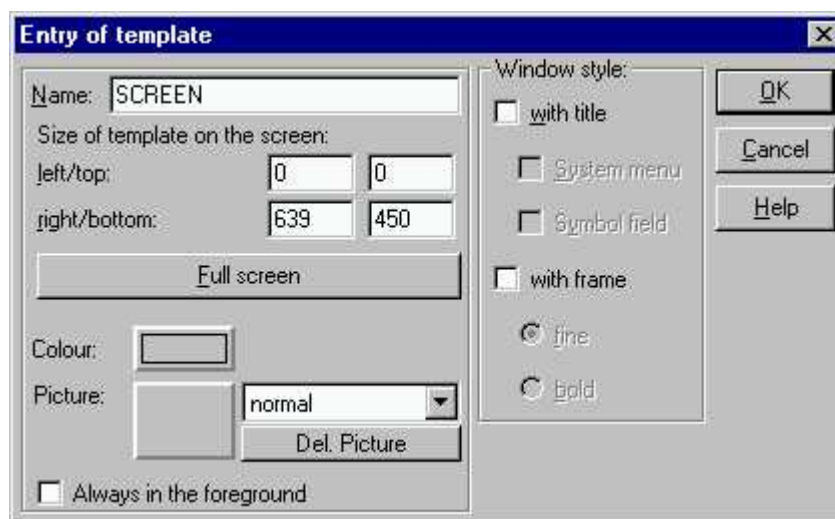
- ▶ "RT changeable data" tab:  
During the first project download, ensure that all checkboxes are deactivated:



## Creation of templates and pictures

Templates are used as the basis for the creation of pictures.

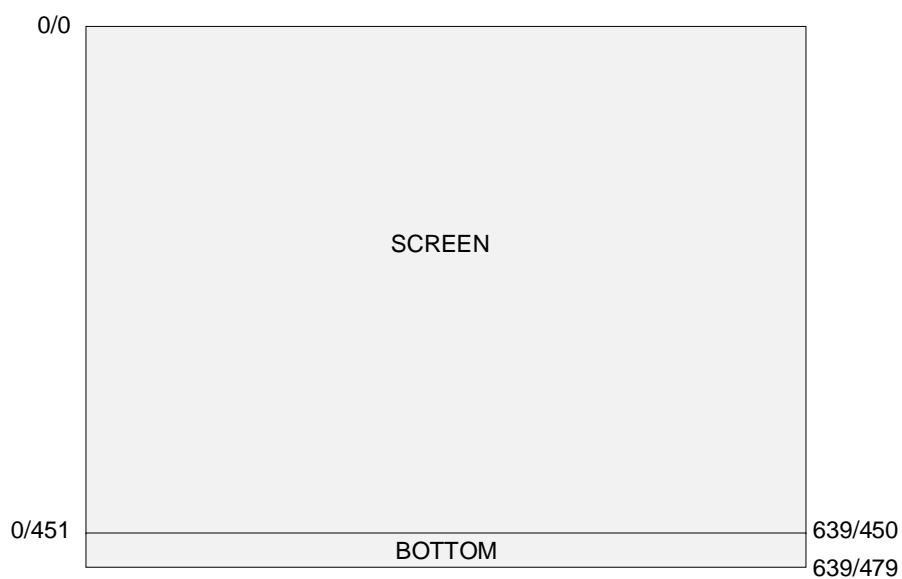
- Pictures -> Templates -> "Entry of template"  
Example: SCREEN template (e.g. shown with VGA resolution here):



Create a further template with the name BOTTOM and the following size in the same way:

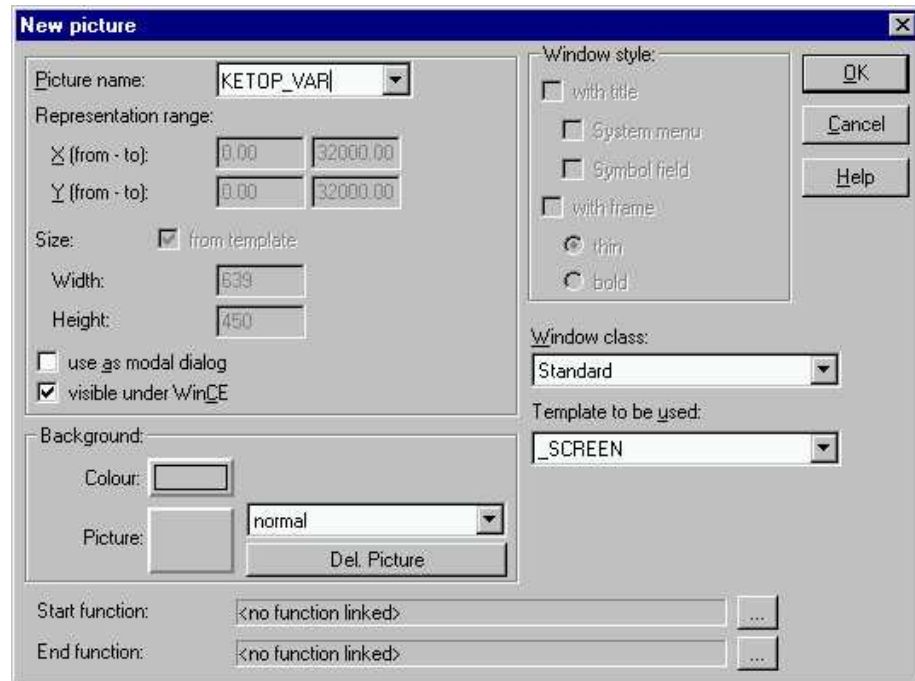
left/above: 0/451 and  
right/under: 639/479

The specified template appears as follows:



For the creation of pictures, proceed as follows:

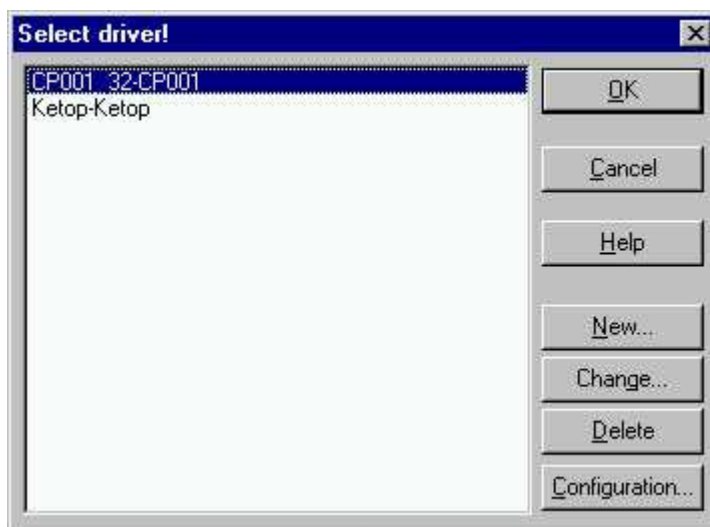
- ▶ Picture -> New picture... -> "New picture"  
Specify a picture name (e.g. KETOP\_VAR) and choose a template on which the drawing should be based (e.g.: SCREEN).



## Driver selection

Select the required driver and its configuration here.

- ▶ Tools -> select driver -> "select driver"



## KEBA-specific zenOn driver

The KEBA specific drivers are described below. The descriptions of all other drivers are given in the description from the COPA-DATA company.

Before using the CP001 or KeTop driver in zenOn, please check for the actual versions.

Verify the version dates of the existing drivers on your PC with the version dates of the drivers on the Startkit CD (\ketop\_cb23x\drivers\zenOn\). You can also check the website [www.copadata.at](http://www.copadata.at) (support -> download) for a new driver version.

When the existing files are older than these files then proceed as follows.

- ▶ Copy \*.exe files to following directory on your PC:  
...\zenon550\
- ▶ Copy \*.dll files to following directory on your PC:  
...\zenon550\ce\ce300\_arm\

### CP001

CP001 is a communication driver that is based on a KEBA-specific serial protocol.

#### HARDWARE:

The CP001 driver is used to connect the KEBA KeTop T50 and T100 handheld terminals to various bus types and controls by means of the Gateway.

Functionality:

The driver fetches cyclical values from the target control. In this case, a data area with a configurable size is written and read during each request. The data types that are supported are described in the table further below.

Prerequisites:

<i>Hardware PC</i>	<i>Software PC</i>	<i>Hardware control</i>	<i>Interface</i>	<i>Protocol</i>
RS232 or RS422 serial interface	-	Gateway according to the bus. Control with expander program.	COM1-COM64	CP001

Software installation:

PC:

Copy the `CP001_32.EXE` driver file into the current zenOn directory (if not already available). Under CE, the `CD001_CE.D11` file must be in the RT directory.

zenOn.ini entries:

None

Definition of the driver in zenOn:

See "General driver documentation" from COPA-DATA.

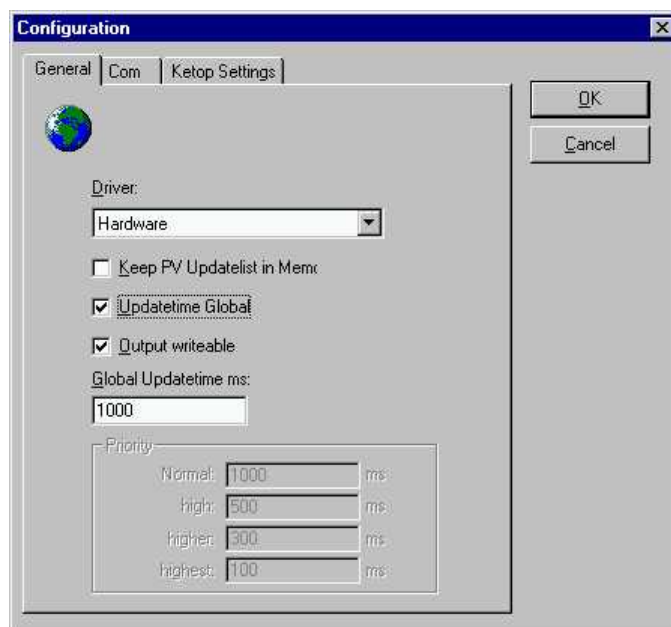
Limitations:

None known.

DRIVER CONFIGURATION:

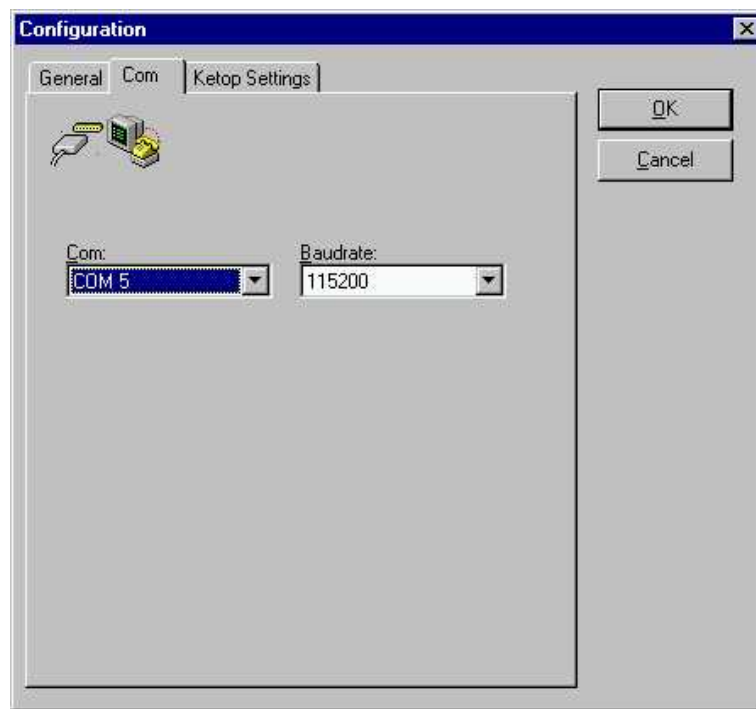
General information:

See "General driver documentation" from COPA-DATA.

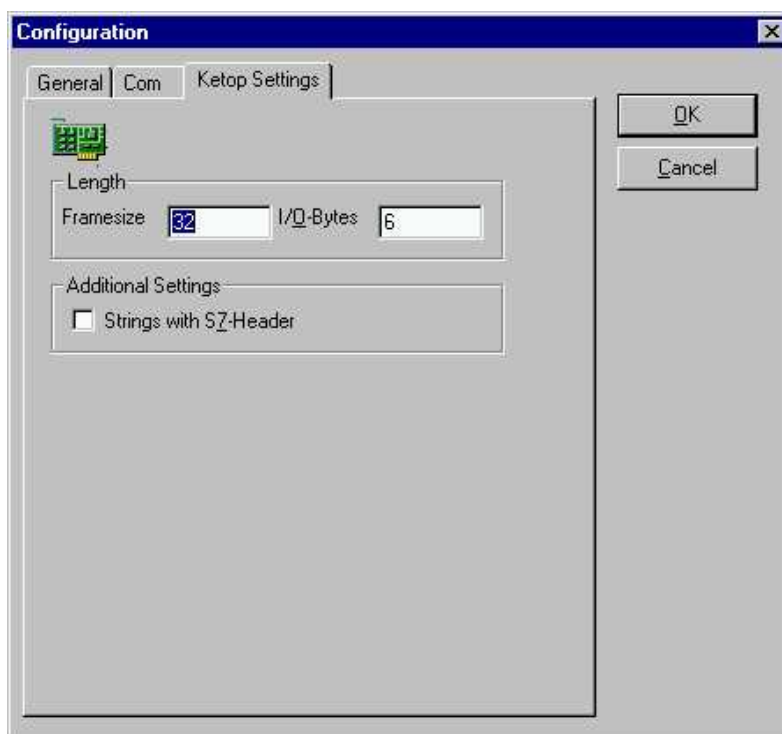


Keep PV Updatelist in Memory	<p><b>Checkbox activated:</b> All project engineered variables, independent of whether they are displayed or not, are constantly updated with the selected update time. Thus, the whole current process map is held in memory. NOTE: This increases the loading of the bus.</p> <p><b>Checkbox deactivated:</b> Only the variables displayed on exactly one image are updated.</p>
Updatetime Global	<p><b>Checkbox activated:</b> Each project engineered variable is re-read or updated with the selected time.</p> <p><b>Checkbox deactivated:</b> The individual variables can be assigned 4 priority levels (normal, increased, high, highest) with respect to the update time. The times for the individual priority levels can be set under "Priority".</p>
Output writeable	<p><b>Checkbox activated:</b> Always active.</p> <p><b>Checkbox deactivated:</b> Not possible.</p>
Global Updatetime ms	Minimum time input: 100 ms





Com:	Selection of the KeTop COM ports
Baudrate:	Selection of the Baud rate with which the KeTop communicates.



Framesize:	Defines the data length on the bus (header + I/O bytes).
I/O-Bytes:	Number of direct data which are always written and read in bytes. Directly predefines the maximum offset for the I/O data.
Strings with S7-Header	If this checkbox is activated, S7-conform strings are expected. These strings have a maximum length of the string in the 1st byte, and the current length of the string in the 2nd byte and then the string data.

## PROCESS VARIABLES

Example of all possible zenOn data types:

PLC	zenOn
I8	i/u8bit (with preceding sign)
I16	i/u16bit (with preceding sign)
I32	i/u32bit (with preceding sign)
U8	i/u8bit
U16	i/u16bit
U32	i/u32bit
F32	float32
Boolean	Boolean
String	String

Variable definition:

The assignment is made in bytes.

I (Input)	Bits, bytes, words, Dwords,... I (Input) bytes are read from KeTop.
O (Output)	Bits, bytes, words, Dwords,... O (Output) bytes are written from KeTop.

Those variables that are transmitted in each packet (directly after the header) in a fixed order which is always the same are denoted as I/O bytes. The order is specified by the allocation of offsets.

## Example:

Offset 0 = 1st byte  
Offset 1 = 2nd byte  
Offset 2 = nth byte  
Offset 3 = n+1th byte

Data	Bits, Bytes, Words, DWords, String
------	------------------------------------

Data bytes are each data that can be cyclically read or written (with adjustable update time) block-by-block. In contrast to the I/O bytes, data bytes are not necessarily sent with every packet (depending on the packet length and the number of projected data bytes). See the following example:

Configuration of the CP001 driver  
Frame size: 32 bytes  
No. of I/O bytes: 3 bytes

Number of projected data bytes: 40 (data byte 0 up to data byte 39)

Since the header of a packet is 6 bytes long, in a 32-byte long packet only 26 bytes are available for the data transport.

Furthermore, in this example, another 3 I/O bytes are projected, whereby only 23 data bytes can be transmitted per packet.



In this example, the KeTop would require 2 read requests to be able to update all 40 data bytes:

1. Packet: READ REQUEST of data bytes 0 - 22
2. Packet: READ REQUEST of data bytes 23 - 39

POSSIBLE ERROR SOURCES:

## Checklist:

- Has the “Error – text file“ been analyzed?  
Which errors have occurred?
- The zenOn project is sent to [support@copadata.at](mailto:support@copadata.at)

## Error messages:

Error number:	Explanation
x1001	Too many repeats
x1005	Packet length incorrect.
x1007	Interface cannot be opened.
x100B	Packet does not begin with an STX.
x100C	Offset for I data area too large.
x100D	Offset for O data area too large.
x100E	No. of read I data does not correspond to the project engineering.
x100F	The command contained in the packet is not recognized or is not supported.
x1010	Serial interface cannot be closed.
x1011	No response data received.
x1012	ACK not received.
x1013	ETX not available in the response data.
X1014	Data when reading too long for bus width. Set project planning correctly.
X1015	Data when writing too long for bus width. Set project planning correctly.

## Interface logging:

For this driver, the logging function can be activated by creating the section  
[RS232LOG]

LOGCOMx=1

in the Projekt.INI. If logging is activated, all data which are exchanged via the serial interface are written to the LOG\_COMxxx.txt file in the directory of the driver. X must be replaced by the number of the interface used. As a considerable data amount is recorded in this case, the logging should only be activated if a fault occurs and then only for a short period of time. This file is also created under CE. To avoid damage to the memory card, the logging function should only be activated for a short time.

### Priority of communication:

In the project INI, it is possible to influence the priority of the communication thread.

This setting can be made by creating the following section.

```
[COMMUNICATION]
```

```
PRIORITY=0 (-15...+15)
```

PRIORITY defines the priority of the communication thread.

In this case, 0 is normal priority or the default setting. Positive values up to max. 15 increase the priority, negative values to min. -15 reduce the priority

## KeTop

The KeTop driver provides the KeTop-specific functions (hand wheel, potentiometer, keyboard) under zenOn.

To transmit these values to a control, the KeTop driver variants must be linked with the communication driver variants (creation of assignments).

### HARDWARE:

Requirements:

<i>Hardware PC</i>	<i>Cable</i>	<i>Hardware control</i>	<i>Interface</i>	<i>Protocol</i>
KEBA KeTop CE terminal		KEBA KeTop onboard peripheral		

### Connection

Direct internal connection via `KetopAPI.DLL`.

Software installation:

#### PC:

Copy the `KETOP.EXE` driver file into the current zenOn directory (if not already available) and enter it in the `TREIBER_DE.XML` file via the tool 'TreiberInfo.exe'.

#### Control:

Copy the `KETOP.DLL` driver and compatible `KetopAPI.DLL` into the zenOn directory of the CE terminal.

zenOn.ini entries:

None

Project.ini entries:

None

Definition of the driver in zenOn:

See "General driver documentation" from the COPA-DATA company.

DRIVER CONFIGURATION:

General:

See “General driver documentation“ of the COPA-DATA company.

PROCESS VARIABLES

Objects for process variables in zenOn:

Object	Read	Write
HANDWHEEL	y	Y
LED	y	Y
ANALOG	y	N
LED-BIT	y	Y
KEY	y	N

HANDWHEEL (word):

The word contains the current counter reading of the handwheel. The counter reading can also be set. The current handwheel value is cyclically polled with the selected update time.

**LED (byte):**

The byte contains the current status of an LED. The status can also be set.  
The settings have the following meanings: 0 Off, 1 On, 2 Flashing.  
Offset: Number of the LED (1..7)

**ANALOG (byte):**

The byte contains the current status of an analog channel.  
Offset: Number of the analog channel (1=potentiometer, 2=joystick-X, 3=joystick-Y)  
The values of the analog channel are cyclically polled with the selected update time.

**LED-BIT (bit):**

The bit contains the current status of an LED. The status can also be set.  
The settings have the following meanings: 0 Off, 1 On  
Offset: Number of the LED (1..7)

**KEY (bit):**

The bit contains the current status of the virtual key with the key code of the offset. 0 means not pressed, 1 means pressed  
Offset: Key code of the key (0..255)  
Status changes of the KE (bit) are transmitted immediately (spontaneously).

**POSSIBLE ERROR SOURCES:****Checklist:**

- Is the `KeTopAPI.DLL` on the terminal?
- Are the offsets correctly set?

**Error codes:**

At present, the driver does not write an error log file.

**Interface logging:**

At present, the driver does not support any logging.



## Project download from the programming PC to the KeTop

To load the zenOn project previously created onto the KeTop, proceed as follows:

- ▶ 1.) Establish a hardware connection between the KeTop and programming PC. Selection can be made between an Ethernet or RS 232 C connection (by means of the XD040 KeTop download cable).
- ▶ 2.) Start the transport service on the KeTop.  
If this was not already started after acknowledgement of the time input window after powering up, start the transport service manually by calling up the file \IPSM\sysssrvce.exe.
- ▶ 3.) Ensure that the parameters set under  
File -> Project configuration -> Project -> Configuration project;  
"Remote transport" tab agree with the parameters of the transport service and the hardware connection.
- ▶ 4.) Set up the connection between the programming PC and KeTop:  
Select the "Establishing remote connection" button in the zenOn editor. A prompt then appears for input of the license serial number of the zenOn-CE runtime. This serial number appears on a sticker in the cable entrance area of the KeTop.

### Notice

- *The input of the serial number is only required for the first project download on the KeTop.*

- ▶ 5.) To start the project download, select the button "Remote transport all project files".
- ▶ 6.) After the successful project download, the start project must be set. To do so, use the button "Set remote Runtime start project".
- ▶ 7.) The project is now located on the KeTop and can be started by use of the button "Start Remote Runtime" or by calling \IPSM\zenOnrce.exe.

### NOTICE:

If the project is already present on the KeTop, it will be automatically started after acknowledgement of the time input window.

## Tips for project planning

### Changing the runtime language

The runtime language affects the menus, input window, error messages etc. amongst others.

The runtime language can be changed over by changing the entry in the file `\IPSM\zenOn.ini`.

The following language DLLs are available on the KeTop:

Language	DLL	Ini entry (LANGUAGE= )
German	rgermace.dll	german
English	renglice.dll	english
French	rfrancce.dll	french
Italian	ritalice.dll	italian
Russian	rrussice.dll	russian
Spanish	rspanice.dll	spanish

### Double-click problems

If you would like actuate soft buttons repeatedly one after the other, it is necessary to deactivate the double-click functionality in Windows CE. It is possible that not all key actions get through to the control.

### Bitmaps

- Bitmaps are not scalable in zenOn, i.e. they must already know before the project planning for which display resolution the project is to be created.
- Bitmaps that are too large will adversely affect the performance of the KeTop.

### Reaction times

The reaction time is dependent on the following factors amongst others:

- Baud rate
- Data length: In the case of the CP001 driver, the data length should be defined with an appropriate value. i.e. no more data bytes than absolutely necessary should be transmitted.
- Update rate:  
An unnecessarily high global update time loads the system and should, thus, also be defined with an appropriate value.

## Variables

- During the project planning, it should be noted that the variables used lie together in a block. As a result, the reading and writing of the variables can be carried out block-by-block, which contributes to an improvement of the performance.
- In the case of the CP001 driver, it is possible to define variables that must be regularly transmitted as I/O data.  
These variables are transmitted along with every packet that is transmitted between the control and KeTop.
- The number of variables on a page has a direct effect on performance. Depending on the number of variables to be read in, the times for changing the picture can be distinctly different.  
The "Keep PV Updatelist in Memory" function can in fact reduce the times for changing the picture to a minimum, but, however, regular reading of all variables in the project causes a high extent of utilization of the interface.
- Dummy variables:  
If, during project planning, only I/O bytes should be used, at least one data byte variable must be planned, however, so that it is possible to carry out cyclical data exchange.  
Procedure:  
Plan one variable and mark "Keep PV Updatelist in Memory" with a cross. This variable must not be displayed on a picture.

## Remote Software ActiveSync

This software is used as a debug interface and to adjust and transmit data from and to a PC.

The remote software ActiveSync is a product by Microsoft and can be downloaded free of charge from the following download address:

<http://www.microsoft.com/mobile/pocketpc/downloads/activesync35.asp>

The data connection from the KeTop to the PC is established as follows:

- ▶ 1. Remove the cover of the cable entrance area on the KeTop.
- ▶ 2. Plug in the Download Cable KeTop XD 040
- ▶ 3. Start the ActiveSync software on the PC
  - a) select 'File' -> 'Delete Partnership' ...
  - b) deactivate '( ) Allow serial cable or ... ' -> <OK>
  - c) activate '(x) Allow serial cable or ... ' -> <OK>
- ▶ 4. Start the ActiveSync-Software on the KeTop:  
*Start -> Programs -> Communication -> Active Sync*
- ▶ 5. At the PC:
  - Connect KeTop as a "Guest" (Set **Up a Partnership -> No**).

Now the data connection between the KeTop and the PC is established. In case of communication problems, check the following conditions:

- Max. time delay between step 3c) and step 4. = 30s.
- For further attempts repeat steps 3. to 5.
- The ActiveSync software must be started manually.  
(Plugging in the connection cable does not start ActiveSync!)
- The factory-set baudrate for the remote connection is 115 kBaud.  
In case of communication problems the baudrate can be reduced.  
The baudrate can be changed as follows:

- Start -> Programs -> Communication -> Remote Networking
- select "com1\_115k"
- File -> Properties
- SP1 on COM1: -> Configure -> Port Settings  
(...change Baud Rate...)
- Close all Windows with <OK>

(optional: Start -> Programs -> KeTop -> Registry Backup -> <OK>)

## 10 KeTop - Specific Operating Instructions

This chapter describes the KeTop-specific settings and the differences to standard Windows CE devices.

### Notice

- *The date and the time are not stored in the KeTop and, if needed, must be set anew after turning off/on. The date and the time may be very important for the log data for example.*

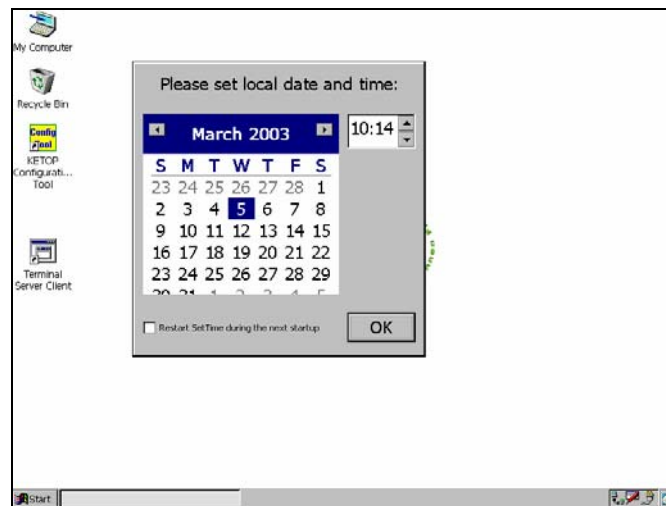
### Notice

- *At settings that have not been made with the ConfigTool, the modifications must be saved as follows:*

*Start -> Programs -> KeTop -> Registry Backup  
(or in the explorer with the command „\windows\regflush.exe“)*

## Setting of Date and Time

During start-up the KeTop shows automatically the window for setting date and time:



Should the window for date and time not be shown, the checkbox has to be deactivated. This must be saved in the registry ("Start -> Programs -> KeTop -> Registry Backup").

The KeTop is not equipped with a real time clock. So, if you use the time it must be set during each star-up.

## KeTop Configuration Tool (ConfigTool)

The ConfigTool is used to calibrate the operating elements, to control the functionality of the operating elements, to select the Ethernet interface and to set start-up functions.

To activate the tool, select the following item:

*Start -> Programs -> KeTop -> KeTop Configuration Tool*  
(or by double-clicking on the file „Configuration\_Tool.exe“ in the Windows directory).

### *Notice*

- *The appearance of the Config tool depends on the operating elements installed in KeTop (the menus for override potentiometer, joystick and handwheel are only displayed when the operating elements are installed in the device).*

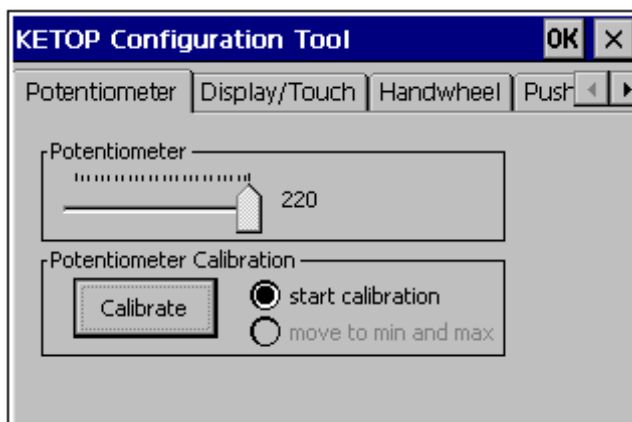
After carrying out modifications on the configuration and quitting the program, you can save all modifications in the registry. A window with the following message and options appears:

*Data has changed! Do you want to write the registry to the flash? Yes/No*

With „No“, the settings are contained in the DRAM and will not be stored therefore. They will get lost when the KeTop is turned off.

„Yes“ saves the calibration in the flash. The settings are preserved after a restart of the device. The device is blocked during the saving procedure (takes approx. 10 s).

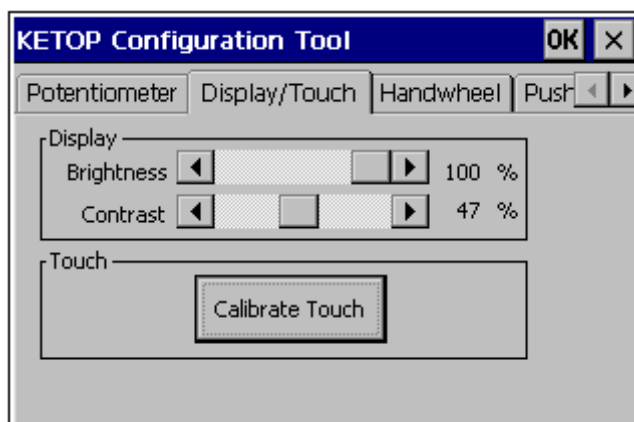
## Calibration of Potentiometer



A number is displayed next to the slide bar. During calibration, this number specifies the current value of the ADC (value range: 0..4096) and only controls the function. After calibration, the number specifies the current value of the potentiometer (value range: 0 ...127). This value is also the value the KeTop API function *KtpGetOverridePoti* supplies to the application.

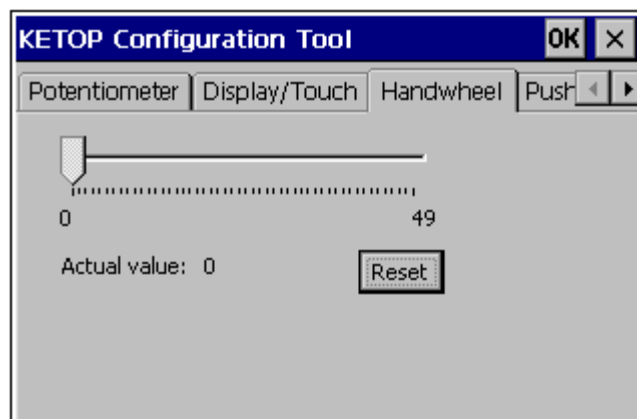
## Display and Touch Screen Settings

In this window, the brightness and the contrast for the display are set.



In this window you can also calibrate the touch screen.

## Setting the Handwheel to Zero

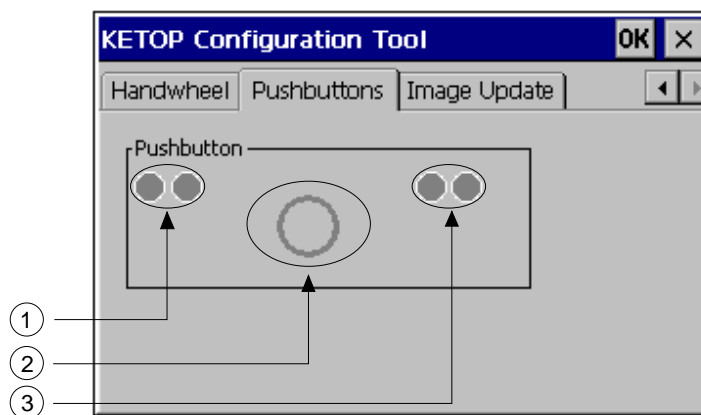


The handwheel outputs a 16-bit value which can be processed in the customer application as needed (e.g.: 0-65535, +-32767, ...). In the dialog box, the value from 0 to 65535 is displayed. When the device is turned on, the value of the handwheel will be set to 0. If the key „Set“ is pressed, the current value of the handwheel will be replaced by the start value.



## Control of Buttons (Pushbuttons)

This toolbox enables a simple functional control of the buttons of the KeTop.



- 1.....Symbols for the left LEDs of the membrane keypad (if available)
- 2.....Symbols for operating elements
- 3.....Symbols for the right LEDs of the membrane keypad (if available)

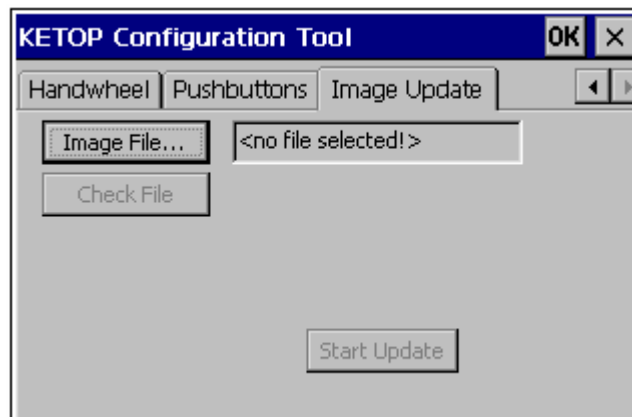
By clicking on the symbol for operating element (2), the corresponding LED of the operating element can be switched on, off or to flashing. The first click switches the LED on, the second click switches the LED to flashing, and the third click switches the LED off again.

The same applies to the LEDs of the membrane keypad (1 and 3).

The actuation of the operating element is signaled by a black ring (a). This ring is grey if the operating element is not actuated.

## Loading an Image File

This toolbox enables loading an image file into the KeTop.



**Image File** Clicking this button enables you to select a new WinCE image file in the file selection dialog.

**NOTICE:**

In the event of an image update, the registry will be deleted and the KeTop rebooted automatically. Following that, the KeTop reboots with the factory settings, and all settings (touch screen, joystick, override potentiometer, system control,...) must be stored anew.

**Check File** The selected file can be checked for validity. "Check File" will only be active if a valid image file has been selected.

**Start Update** This button starts the update. The progress status is displayed in the bar below. "Start Update" will only be active if the selected image file has been successfully checked for validity (with "Check File").

## Checking the Operating and Control Elements

For checking the functionality of the operating and control elements of the KeTop, the KeTop provides the software „Systemcheck“. This software is contained in the start menu and can be operated intuitively. It is called as follows:

*Start -> Programs -> KeTop -> „System Check“*

Provided they exist on the device, the following operating and control elements can be checked:

- Override potentiometer
- Electronic handwheel
- Status LEDs
- Membrane keyboard
- Touch screen
- Display
- Illuminated push button
- Key switch
- Selector switch
- IPSM flash file system

The following control elements CANNOT be checked since their internal evaluation via the electronics is not provided:

- Emergency stop switch
- Enabling switches

Moreover the software „Systemcheck“ enables checking KeTop device data, such as CPU memory, interface module, operating elements, display, etc.

## Installation of Programs

Programs can be only installed in the IPSM (Intel Persistent Storage Manager) directory. Since only this directory is stored in the flash, data installed in another way will get lost after turning off the KeTop.

Installation data entered in the registry can be saved with the command „\Windwos\Regflush.exe“.

### *Notice*

- *System files, which are copied to the directory \windows while installation, do not remain durably stored. For this reason these files must be copied manually into the directory \IPSM\windows (if necessary, this directory must be created by the user). With a restart of Windows CE these files are copied automatically into the directory \windows. So, these files are again available for the operating system and are again present also after a restart (KeTop T100: valid from OEM Build Vers. 1.4a; KeTop T50: valid from OEM Build Vers. 1.2).*

## Saving Files

Files must be saved in the IPSM directory. The contents of other directories are not saved when the device is turned off.

## 11 Options

This chapter describes the options the KeTop is available with.

### Override Potentiometer

If available, the override potentiometer in the KeTop is evaluated by the software and can be read via the KeTop-API using a program.

The override potentiometer can be used for example for setting the number of spindle revolutions and the forward feed on machine tools.

- Resolution: 0 – 127, linear

### Electronic Handwheel

If available, the electronic handwheel in the KeTop is evaluated by the software and can be read via the KeTop-API using a program.

50 pulses are counted per revolution. Turning the handwheel clockwise increments the count, turning the handwheel counter-clockwise decrements the count (-32768 to +32767, 16-bit value).

#### Main features

- 50 lock-in positions / revolution
- 1 pulse / lock-in position
- 

#### Notice

- *If the KeTop falls to ground, control the position of the handwheel knob. If necessary press down the knob until it snaps in.*

## Illuminated Push-Button

If available, the illuminated push-buttons in the KeTop are evaluated by the software and can be read via the KeTop-API (see page 42) using a program.

The illuminated push-buttons are available as „momentary“ type (labeled 0 or I) or as „maintaining“ type (labeled 0/I).

### WARNING

- After removing the covers of the push-buttons (in case of changing the labelling) take care that the labelling plates are positioned again correctly referring to the functionality of the push buttons.  
(Otherwise the wrong push-button may be used for switching ON or OFF the motor.)

## Key Switch

If available, the key switch is evaluated by the software in the KeTop and can be read via the KeTop-API using a program.

The key switch is available with 3 positions.

## Selector Switch

If available, the selector switch is evaluated by the software in the KeTop and can be read via the KeTop-API using a program.

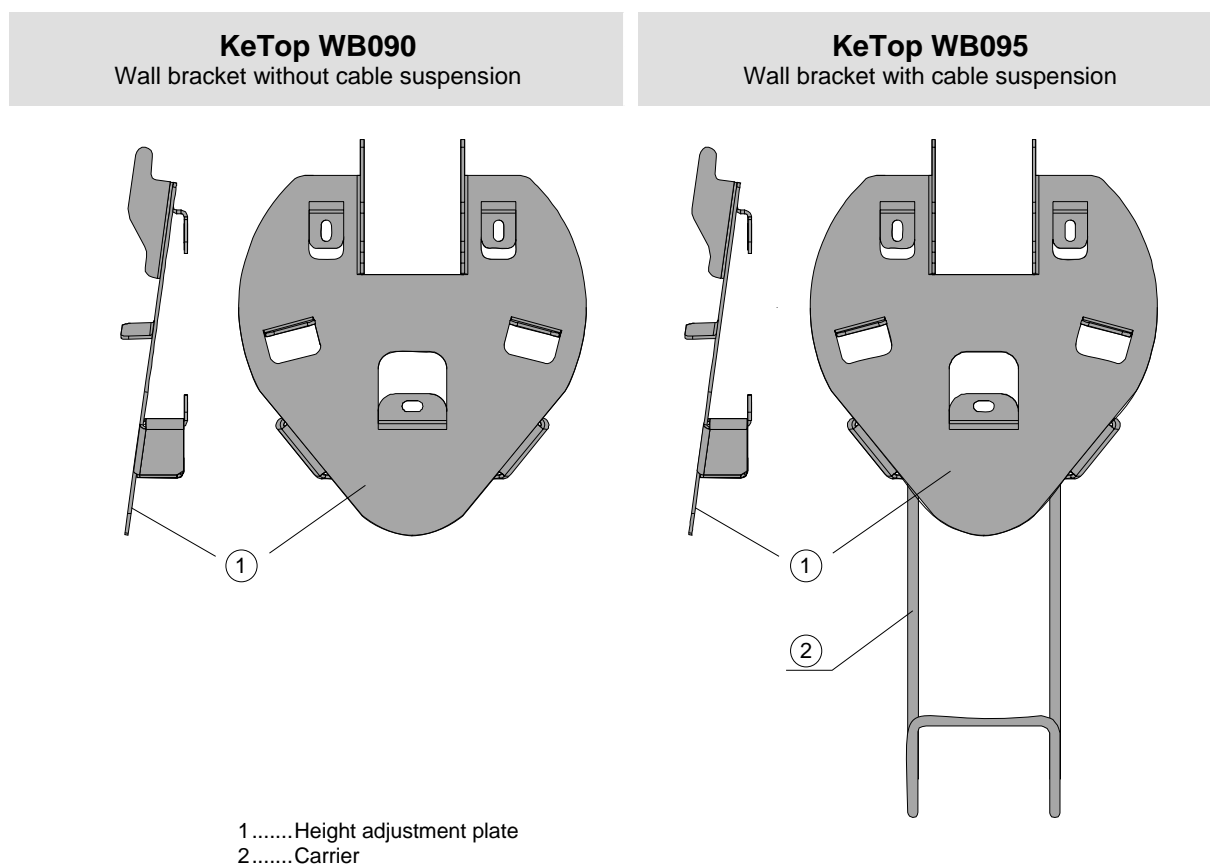
The selector switch is available with 3 positions.

## 12 Accessories

### Wall bracket KeTop WB090 and KeTop WB095

The powder-coated black wall bracket is used for stationary operation or storage of the KeTop.

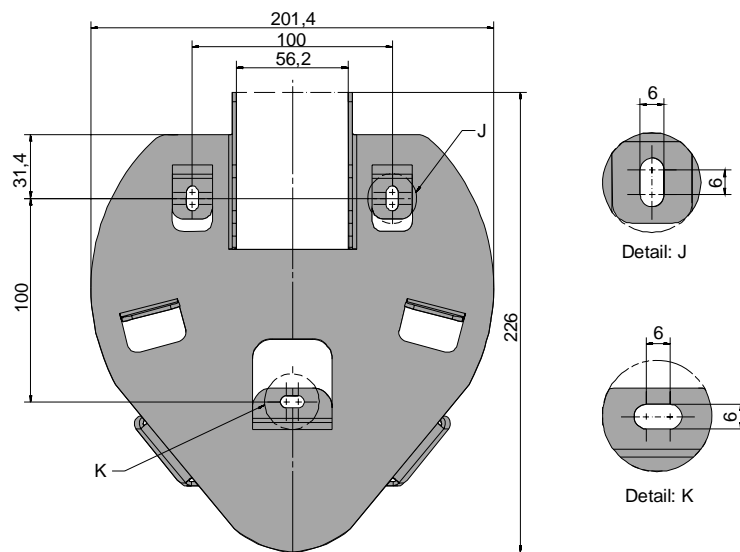
Two types of wall brackets are available:



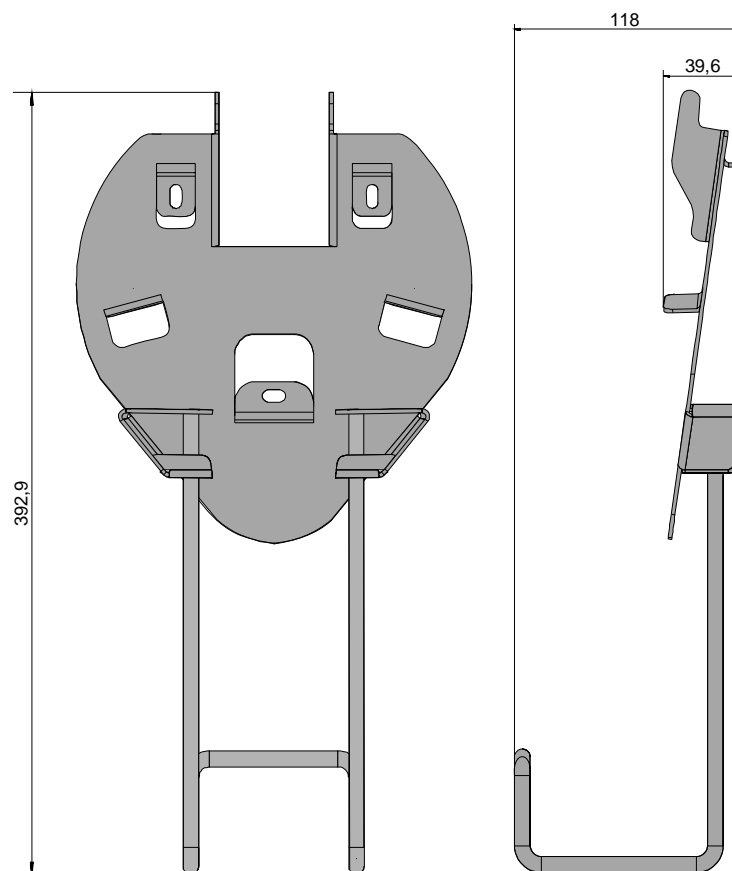

---

Wall bracket KeTop WB100 and KeTop WB110 with KeTop

## Dimensions (mm)



Wall bracket KeTop WB090, front view



Wall bracket KeTop WB095, rear and side view

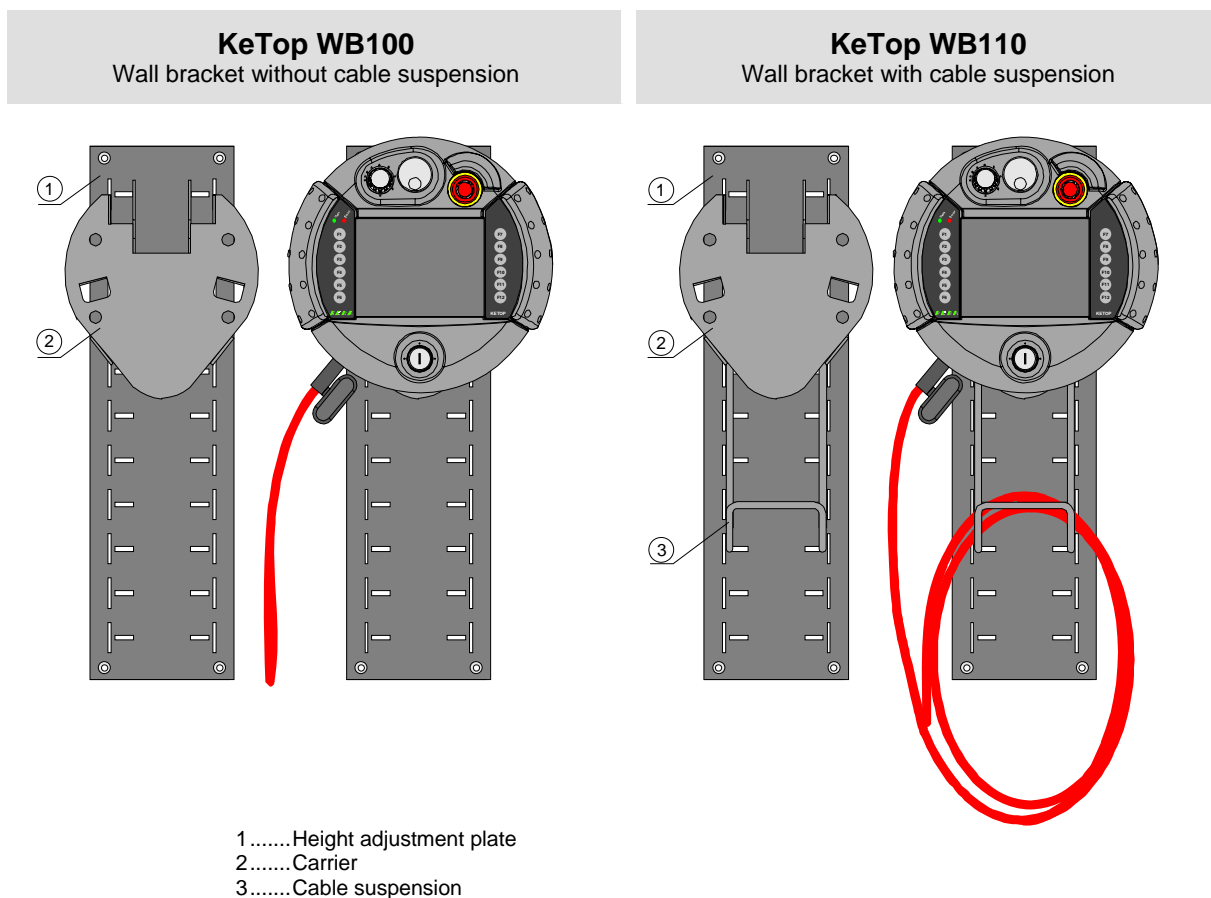


## Wall Bracket with Height Adjustment plate KeTop WBxxx

The powder-coated black wall bracket is used for stationary operation or storage of the KeTop.

The carrier is adjustable in 8 positions over a height of 320 mm (12.6 in). Take care to hang up the carrier in all 4 points in the height adjustment plate. The cable suspension must be mounted on the carrier using the screws delivered with the device.

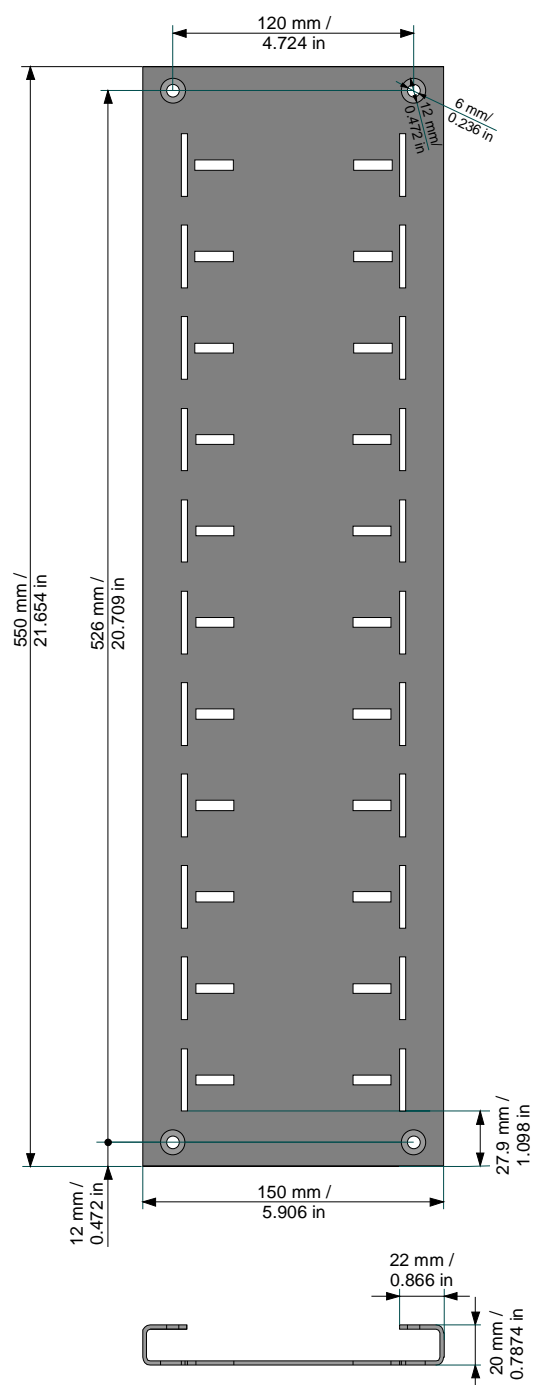
Two types of wall brackets are available:



Wall brackets KeTop WB 100 and KeTop WB 110

## Height adjustment plate

For mounting the height adjustment plate, use suitable screws (not part of delivery).

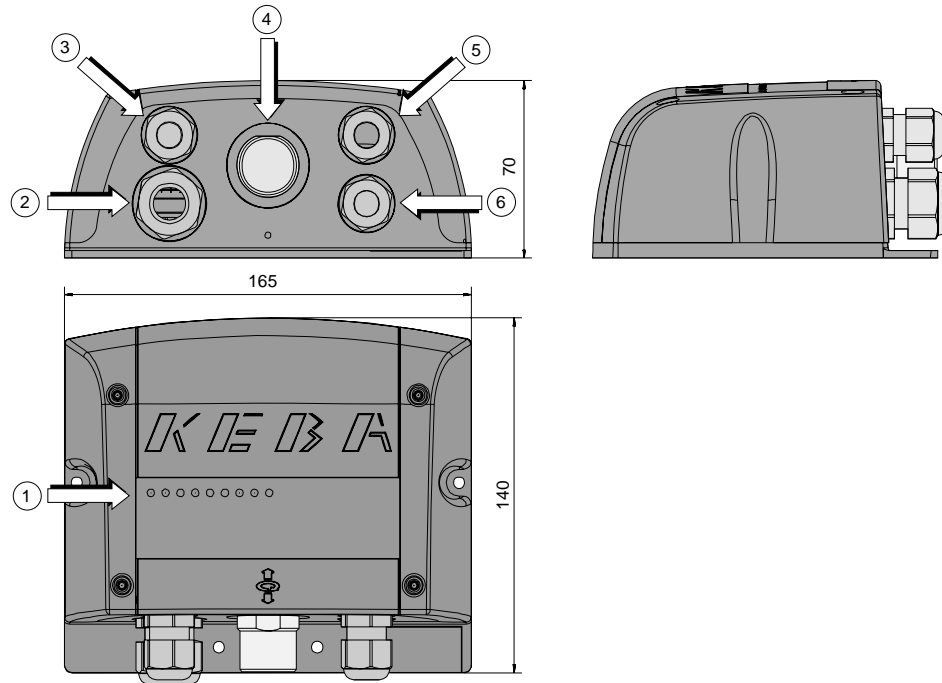


Height adjustment plate for wall bracket WB110

## KeTop CB211 Connection Box

The KeTop CB211 connection box is used for integration of the KeTop in the machine/system. It is suitable for wall mounting and can also be mounted on a mounting rail through the use of the mounting rail assembly kit (KeTop DR200). The KeTop CB211 connection box has the following connections:

### Construction



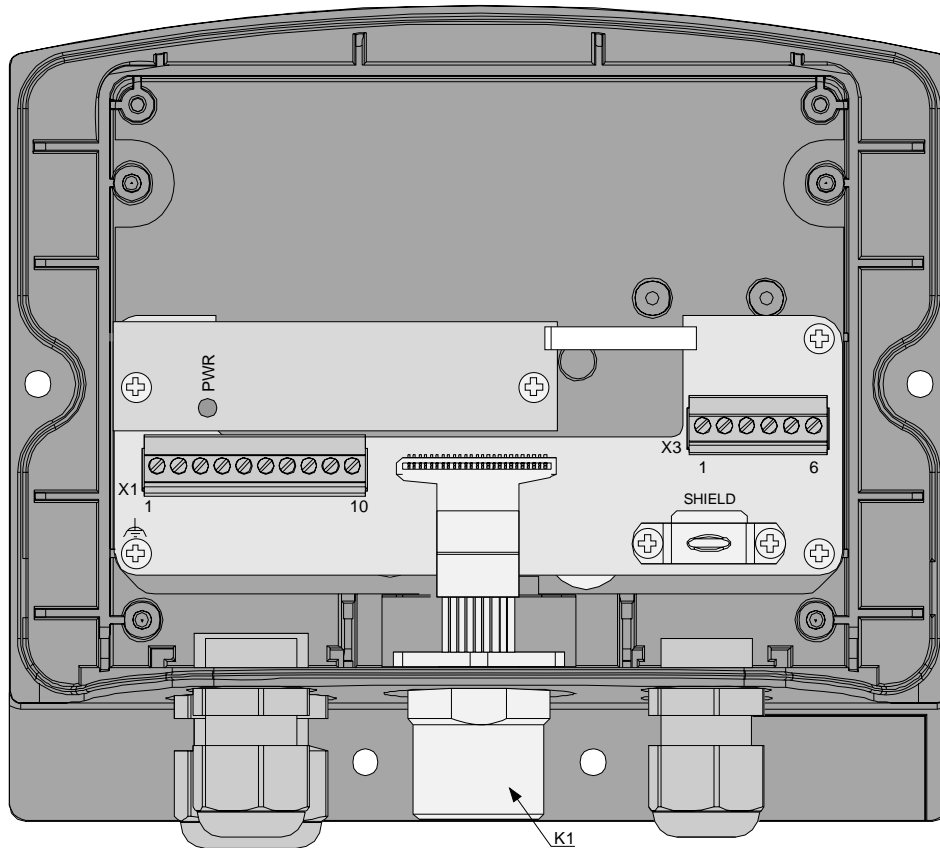
- 1 .....Status and error LEDs
- 2 .....PG gland (M20) for voltage supply, enabling switch and emergency stop
- 3 .....PG gland (M16) for separate functional ground (status as supplied with dummy plugs)
- 4 .....Coninvers female connector for KeTop T100, T50 and T40 - connection cable
- 5 .....PG glands (M16)for data lines
- 6 .....PG glands (M16) for data lines (status as supplied with dummy plugs)

View and device description of the Gateway box

### **⚠ WARNING**

- **The connection box and the handheld terminal meet the safety class III in accordance with EN61131-2 and EN50178.**  
**When connecting the handheld terminal, make sure that all voltages connected to the handheld terminal are safety extra low voltages and isolated from the low voltage supply system by a safety transformer or a similar facility.**

## Interior view



- K1..... 17-pin female connector (Coninvers) for KeTop connection cable  
 X1..... Terminal block for power and control lines (enabling switch and emergency stop)  
 X3..... Terminal block for data lines  
 SHIELD..... Cable shield clamp with connection surface for cable shield of data lines  
 (not used for strain-relief of the cable!)

---

 Interior view of connection box

## Technical data of the connection terminals

The following technical data apply to the X1 and X3 connector terminal blocks already available in the Junction box:

<b>Connection capacity:</b>		
rigid / flexible / wire gages	[mm²]/[mm²]/AWG	0.14-1.5 / 0.14-1.5 / 28-16
flexible with wire end ferrules without / with plastic sleeve	[mm²]	0.25-1.5 / 0.25-0.5
<b>Grid dimension:</b>		3.81
<b>Insulation length:</b>	[mm²]	7
<b>Tightening torque:</b>	[Nm]	0.22-0.25

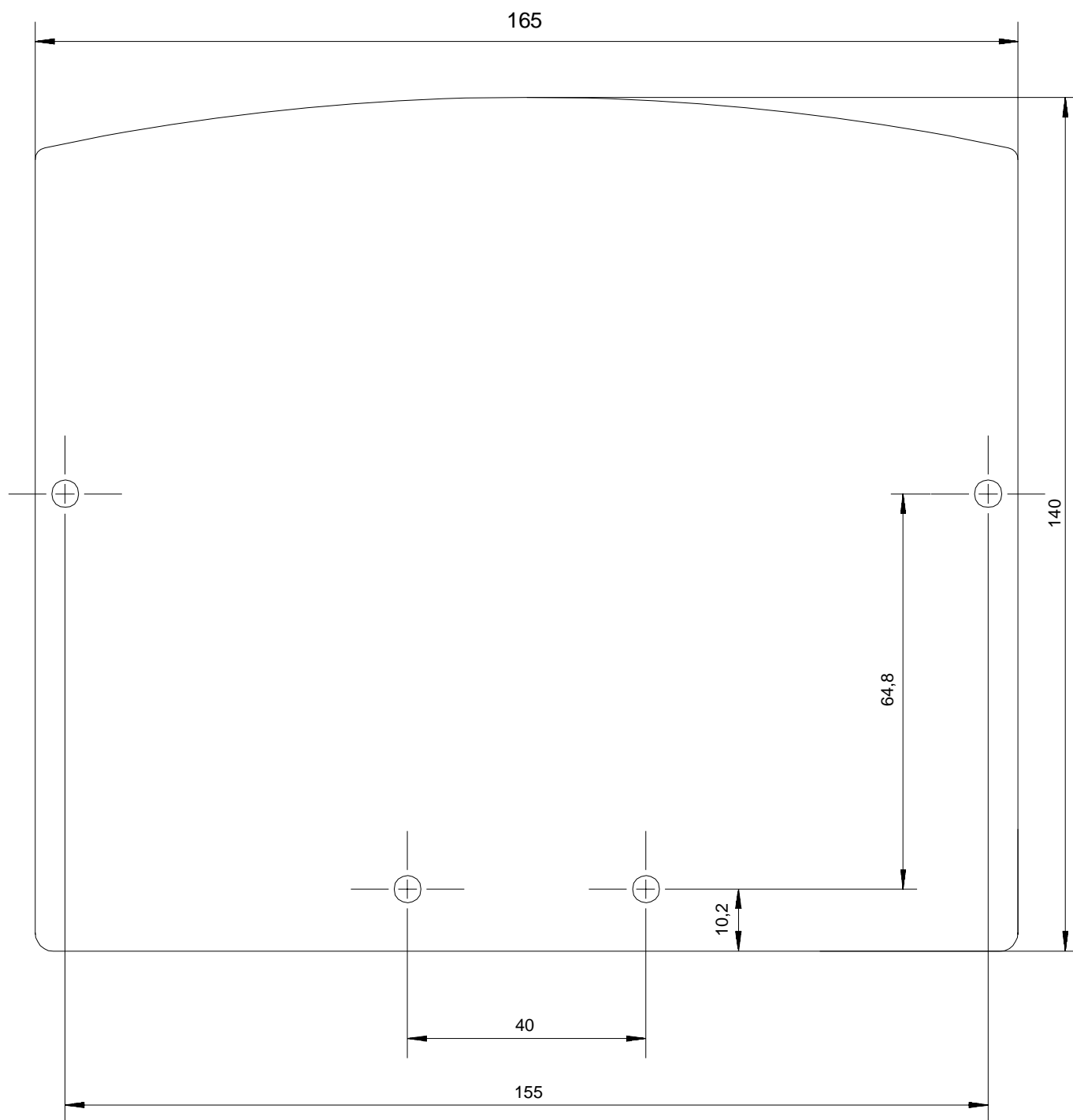
PHOENIX order data:

Gateway-terminal block	PHOENIX Type	Part no.
X1	MCVR 1.5/6-ST-3.81	1827169
X3	MCVR 1.5/7-ST-3.81	1827172
X4, X4B	MCVR 1.5/10-ST-3.81	1827208

### Notice

- Consider the connection capacity of the terminal blocks when selecting the connection cable.
- Use the following screwdriver to connect the wires to the terminal blocks:  
Blade: 0.4 x 2.5 x 80 mm,  
Length: 160 mm
- Multi-line connections (2 wires in one terminal) are not allowed. Use the X4B terminal block for continuing the field bus.

## Drilling template for wall mounting



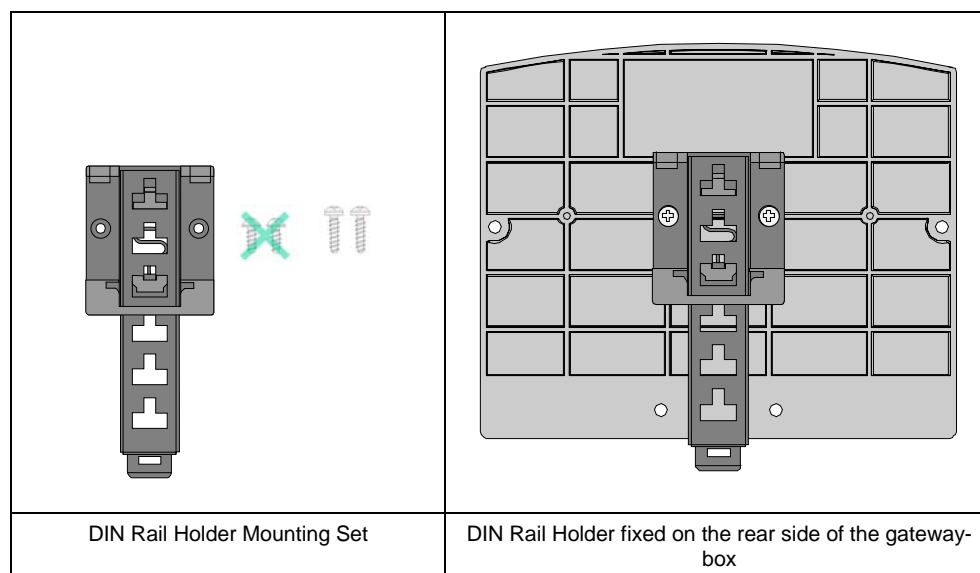
Drilling template for the Gateway box, scale 1:1

For wall mounting, we recommend the following screws and materials:

- Chipboard screw:  $\varnothing$  4 x 40mm  
Head form: flat head  
Max. head diameter:  $\varnothing$  9mm
- Recommended rawlplug:  $\varnothing$  6 x 30mm

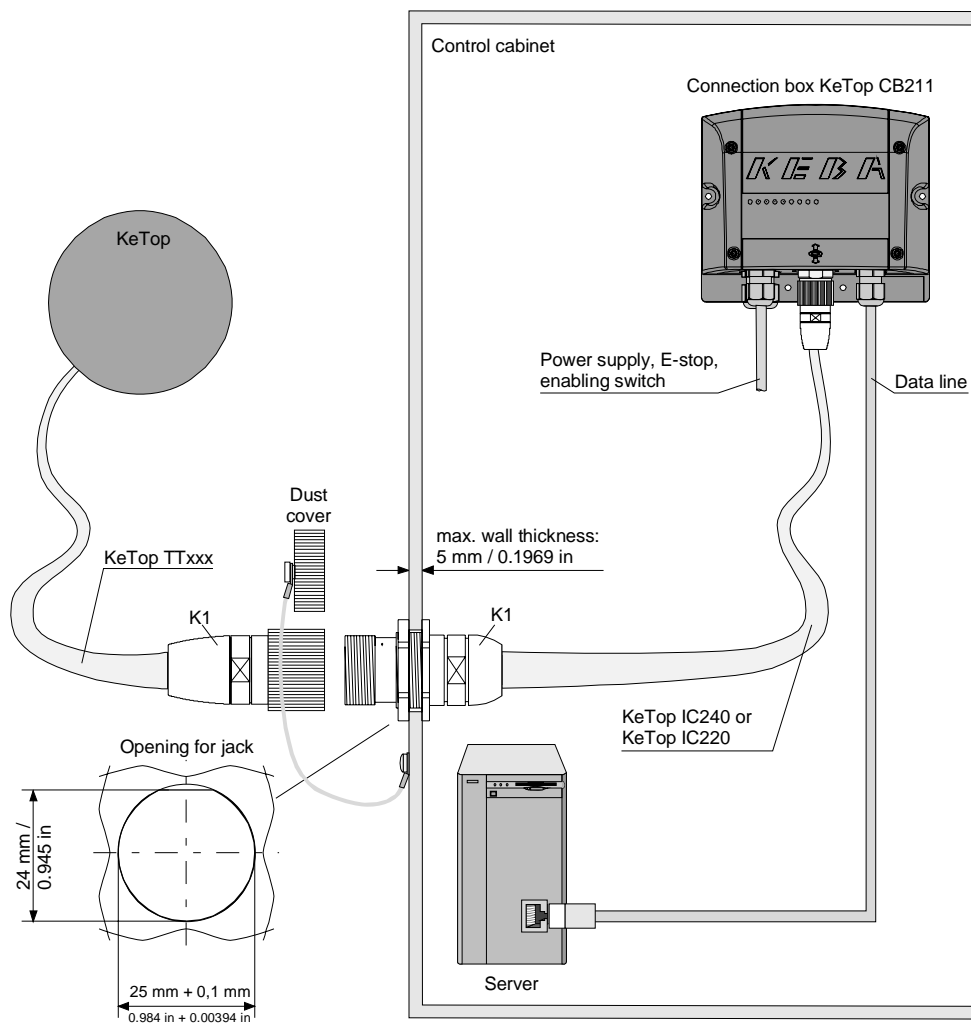
### DIN Rail Holder Mounting Set KeTop DR200

The DIN rail holder KeTop DR200 is available as accessory and will be mounted on the rearside of a KeTop CB2xx gatewaybox. So the gatewaybox can be easily snapped onto a DIN rail.



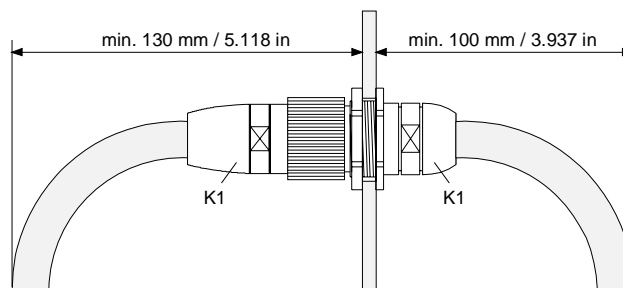
DIN Rail Holder Mounting Set for gatewaybox

## Use of Connection Box



Connection box KeTop CB211 in control cabinet

## Minimum Bending Radius of Cable



Required minimum distance outside and inside the control cabinet



## Technical Data of Connection Box KeTop CB211

### General data

Nominal supply voltage:	24 V DC (Safety extra low voltage)
Supply voltage range:	18 V DC to 32 V DC
Maximum interruption time of supply voltage:	≤ 10 ms (lt. IEC 31131)
Power consumption:	
with KeTop:	3.6 W (200 mA at 18 V DC, 150 mA at 24 V DC)
without KeTop:	10.8 W (600 mA at 18 V DC, 450 mA at 24 VDC)
Inrush current:	max. 5.6 A (with limitation of current)
Safety class:	III in accordance with EN 61131-2 and EN 50178

### Environmental conditions

Operating temperature:	0 °C to 50 °C (32 °F to 122 °F)
Storage temperature:	-20 °C to +70 °C (-4 °F to 158 °F)
Relative humidity (non-condensing):	5 % to 95 %
Vibration resistance (operation):	(IEC 60068-2-6)
	5 ≤ f < 9 Hz 7 mm
	9 ≤ f ≤ 150 Hz 2g (0.0044 pound)
Shock resistance (operation):	15 g (0.033 pound) / 11 ms (IEC 60068-2-27)

### Housing

Construction	
	Double-walled ABS housing
	Withstands grease, oil, lubricants, alcohol, etc.
Flammability class:	UL94-V0
Dimensions:	
Width:	160 mm
Height:	140 mm
Depth:	70 mm
Protection degree:	IP65
Weight:	500 g (1.1 pound)
Display:	Status LEDs

### Accessories

Intermediate cable	Connection box to connection cable
KeTop IC220:	2 m / 6.56 ft
KeTop IC240:	4 m / 13.12 ft
Download cable	
KeTop XD040:	4 m / 13.12 ft. For downloading software and for debugging via S2.
DIN Rail Holder Mounting Set	
KeTop DR200:	For mounting on the rearside of a KeTop CB2xx gatewaybox.

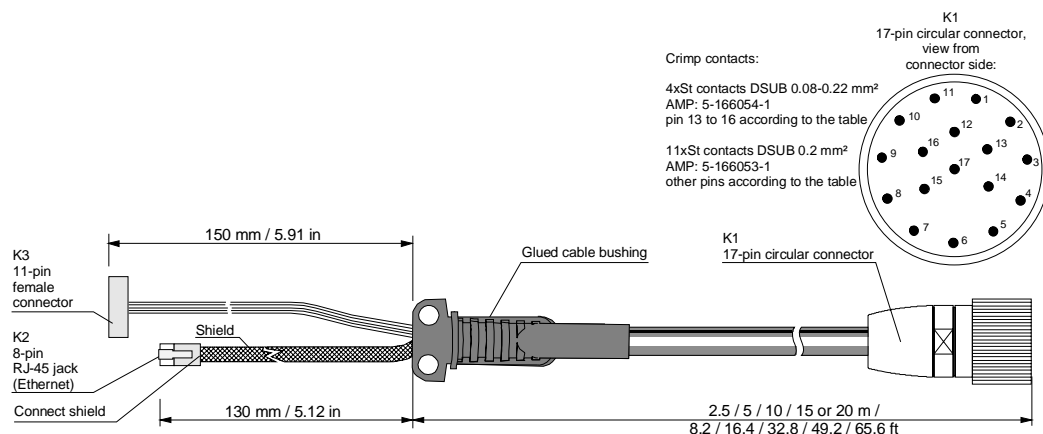
## Connection Cable KeTop TTxxx

The standard KeTop handheld terminals are available with the following cables:

- KeTop TT025 (2,5 m)
- KeTop TT050 (5 m)
- KeTop TT100 (10 m)
- KeTop TT150 (15 m)
- KeTop TT200 (20 m)

The KeTop connection cable withstands water, cleaning agents (alcohol and tensides), oil, drilling oils, grease and lubricants.

Description of signal	K3, 11-pin female connector to S19 on the KeTop	K2, 8-pin RJ-45 jack (S6, RS-422-A)	Connection cable KeTop TTxxx, color of wires		K1, 17-pin male connector, pin No.:
24 VDC	6	-	pink	->	1
GND_IN	7	-	black	->	2
E-stop, circuit 1	8	-	brown-green	->	3
E-stop, circuit 1	9	-	white-green	->	4
E-stop, circuit 2	10	-	grey-pink	->	5
E-stop, circuit 2	11	-	red-blue	->	6
enabling switch, circuit 1, pos.	1	-	brown	->	7
enabling switch, circuit 1, neg.	2	-	yellow	->	8
enabling switch, circuit 2, pos.	3	-	green	->	12
enabling switch, circuit 2, neg.	4	-	grey	->	17
not used	n.c.	-	-	-	9
not used	n.c.	-	-	-	10
not used	5	-	violet	->	11
TD+   CAN+	-	1	blue	->	13
TD-   CAN-	-	2	white	->	14
RD+   SGND	-	3	orange	->	15
RD-   not used	-	6	red	->	16



Connection cable KeTop TTxxx

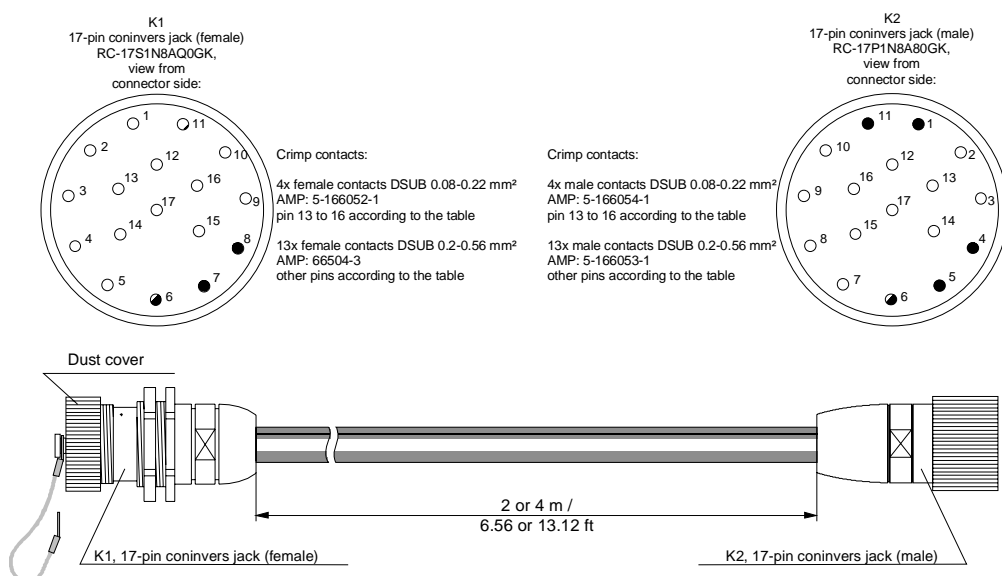
## Intermediate Cable KeTop ICxxx

The KeTop intermediate cable is used to connect the connection box and the Coninvers jack in the wall of the control cabinet. Cables with the following lengths are available:

- KeTop IC020 (2 m)
- KeTop IC040 (4 m)

Description of signal	K1, 17-pin female connector, pin No.:	Intermediate cable ICxxx, color of wires	K2, 17-pin male connector, pin No.:
Enabling switch, circuit 1, pos.	7	brown	<- 7
Enabling switch, circuit 1, neg.	8	yellow	<- 8
Enabling switch, circuit 2, pos.	12	green	<- 12
Enabling switch, circuit 2, neg.	17	grey	<- 17
Not used	10	-	10
24 V DC (+/- 10%)	1	red	<- 1
Not used	9	-	9
Emergency stop, circuit 1	3	green/brown	<- 3
Emergency stop, circuit 1 GND	4	white/grey	<- 4
Emergency stop, circuit 2	5	grey/pink	<- 5
Emergency stop, circuit 2 GND	6	red/blue	<- 6
GND_IN	2	black	<- 2
TD+ (transmit)	13	blue	<- 13
TD- (transmit)	14	white	<- 14
RD+ (receive)	15	orange	<- 15
RD- (receive)	16	red	<- 16
Powerfail	11	violet	<- 11

Shielded signals: TD+, TD-, RD+, RD-



Intermediate cable KeTop ICxxx

## Download Cable KeTop XD040

The download cable XD040 is available with a length of 4 m (13.12 ft) and plugged into the serial port connector S2 in the cable entrance area of the KeTop or into the X6 of the KeTop CB23x Gatewaybox.

The cable is used for downloading software and for debugging.

Handheld Terminal KeTop xxxx		Gateway box KeTop CB23x		Download cable KeTop XD040			PC
Signals	S2 Serial port con- nector	Signals	X6 Serial port con- nec- tor	S1 male con- nector	Description of signals	S2, 9-pin DSUB female connector	Signals (COMx)
-	1*	n.c.	1	1	< - >	1	(ActiveSync)
-	2*	n.c.	2	2	< - >	6	(ActiveSync)
TXD	3	TXD	3	3	< - >	2	RXD
CTS	4	n.c.	4	4	< - >	-	-
RXD	5	RXD	5	5	< - >	3	TXD
RTS	6	n.c.	6	6**	< - >	-	-
-	7*	n.c.	7	7	< - >	4	(ActiveSync)
n.c.	8	n.c.	8	8**	< - >	9	-
GND	9	GND	9	9	< - >	5	GND
GND	10	n.c.	10	10	< - >	-	not used
						7	not used
						8	not used

\*)..... The pins 1, 2 and 7 of the serial port connector S2 are short circuited directly on the CPU board of the KeTop.

If you produce the serial download cable yourself you will need these three wires in the cable for the ActiveSync signal.

\*\*) ..... One pin has been removed from the male connector (prevents from incorrect plugging in).



Download cable KeTop XD040

## 13 Transport Conditions

To avoid damaging the device during further or return transport, the following transport conditions must be fulfilled:

- Always use the original packaging for the transport.
- The environmental conditions for the device (see chapter „Technical Data“) must also be fulfilled during transport.

# 14 Technical Data

## General data

Nominal supply voltage:	24 V DC
Supply voltage range:	18 V DC to 32 V DC
Maximum interruption time of supply voltage:	≤ 10 ms (EN 61131)
Power consumption:	6 W (333 mA at 18 V DC, 250 mA at 24 V DC)
	Inrush current: max. 5.6 A (with limitation of current)
Safety class:	III in accordance with EN 61131-2 and EN 50178

## Environmental conditions

Operating temperature:	0 °C to 50 °C (32 °F to 122 °F)
Storage temperature:	-20 °C to +70 °C (-4 °F to 158 °F)
Relative humidity (non-condensing):	5 % to 95 %
Vibration resistance:	(IEC 60068-2-6)
	5 Hz ≤ f < 9 Hz with 7 mm (0.276 in)
	9 Hz ≤ f < 150 Hz with 2 g (0.0044 pound)
Shock resistance:	25 g / 11 ms (IEC 60068-2-27)

## Housing

Construction	Twin shell ABS housing Withstands grease, oil, lubricants, alcohol, etc.
Flammability class	UL94-V0
Dimensions:	
Diameter:	250 mm / 9.84 in
Total height incl. handle:	125 mm / 4.92 in
Degree of protection:	IP65
Weight:	Approx. 1300 g (with E-stop, without handwheel, without override potentiometer, without cable)
Color display	
Type:	Graphics-capable color STN LC display
Size:	5.7" (120 x 90 mm)
Resolution:	1/4 VGA, 320 x 240 pixels
Representation:	256 colors
Background lighting:	CCFT cold cathode tube (min. lifetime 50,000 hours)
Touch screen:	Analog-resistive
b/w display	
Type:	Graphics-capable STN LC display
Size:	5.7" (120 x 90 mm)
Resolution:	1/4 VGA, 320 x 240 pixels
Representation:	16 grey levels
Background lighting:	CCFT cold cathode tube (min. lifetime 50,000 hours)
Touch screen:	Analog-resistive
Keypad:	
	- Membrane keypad with tactile feedback
	- Keys laid out for thumb operation
	- Left / right-hand operation
	- Customer-specific keypad possible
	- 2 status LEDs
Operating elements:	Two 3-position enabling switches, twin circuit

## Options:

	Override potentiometer
	Electronic handwheel
	Emergency stop switch
1 location for:	Illuminated push-button, momentary 0
	Illuminated push-button, momentary I
	Illuminated push-button, maintaining I/O
	Key switch, 3 positions: I-0-II / 0-I-II

## Processor and interfaces

Processor:	Intel StrongARM SA-1110/206 MHz
Memory:	SDRAM: 16 MB, FLASH: 16 MB
Interfaces:	RS-422-A
	RS-232-C (debug interface in device)
	CF card slot (type I)
Operating system:	Windows CE

## Accessories

Wall bracket	For stationary operation or storage of the KeTop.
KeTop WB090:	Wall bracket without cable suspension.
KeTop WB095:	Wall bracket with cable suspension.
KeTop WB100:	Wall bracket with height adjustment and without cable suspension.
KeTop WB110:	Wall bracket with height adjustment and with cable suspension.
Connection cable	Resistent to twisting, bending and foot traffic, with 17-pin circular conn.
KeTop TT025:	2.5 m / 8.2 ft
KeTop TT050:	5 m / 16.4 ft
KeTop TT100:	10 m / 32.8 ft
KeTop TT150:	15 m / 49.2 ft
KeTop TT200:	20 m / 65.6 ft
Intermediate cable	Connection box to connection cable
KeTop IC220:	2 m / 6.56 ft
KeTop IC240:	4 m / 13.12 ft
Download cable	(not for KeTop T30)
KeTop XD040:	4 m / 13.12 ft. For downloading software and for debugging via S2 (KeTop) or via X6 (KeTop CB23x).
Connection / Gatewax boxes	(KeTop CB23x not for KeTop T30)
KeTop CB211:	Connection box for separating the data and control lines.
KeTop CB230:	Gateway box for connecting to CAN via the RS-422-A interface.
KeTop CB232:	Gateway box for connecting to CANopen via the RS-422-A interface.
KeTop CB233:	Gateway box for connecting to DeviceNET via the RS-422-A interface.
KeTop CB234:	Gateway box for connecting to INTERBUS via the RS-422-A interface.
KeTop CB235:	Gateway box for connecting to PROFIBUS-DP via the RS-422-A.
KeTop CB236:	Gateway box for connecting to PROFIBUS-MPI via RS-422-A.
Visualization software	Only for KeTop T100 and KeTop T50
KeTop PT001:	zenOn 5.50
Start Kit	
KeTop SK001:	Manuals English/German on CD
	SDK for Windows CE
	Programming tool KeTop PS040
	Demoapplication zenOn 5.50
	Demoapplication PLC

## Spare parts

Touch styluses

KeTop E-TP001:

3 original KeTop touch styluses

Service lid

KeTop E-SD050:

Service lid of cable entrance area with seal and attachment material.



## 15 CE conformity and standards

### CE conformity

The CE marking indicates that this KEBA product complies with all applicable directives of the European Community.

For this handheld terminal, these are the EMC directive RL 89/336/EG and the machinery directive RL 98/37/EG.

As defined in the machinery directive, this device is considered to be an safety component.

The standards used for presumption of conformity are contained in the next chapter.

A "declaration of conformity" in agreement with the above-mentioned directives has been delivered and can be viewed at KEBA in Linz, Austria.

### Standards

#### EMC directives

EN 61000-6-4:2001

Electromagnetic compatibility (EMC): Generic standards - Immunity for industrial environments

EN 61000-6-2:2001

Electromagnetic compatibility (EMC): Generic standards - Emission standard for industrial environments

IEC 61131-2: chapters 7,8

(Programmable Controllers  
Part 2: Equipment requirements and test)

#### Machinery directives

EN 292-1:1991

Safety of machinery - Basic concepts, general principles for design - Basic terminology, methodology

EN 292-2:1991 + A1:1995

Safety of machinery - Basic concepts, general principles for design - Part 2: Technical principles and specifications

EN 418:1992

Safety of machinery; emergency stop equipment, functional aspects; principles for design

EN 614-1:1995

Safety of machinery - Ergonomic design principles - Part 1: Terminology and general principles

EN 894-1:1997

Safety of machinery - Ergonomics requirements for the design of displays and control actuators-Part 1: General principles for human interactions with displays

EN 894-2:1997

Safety of machinery - Ergonomics requirements for the design of displays and control actuators - Part 2: Displays

EN 894-3:2000

Safety of machinery - Ergonomics requirements for the design of displays and control actuators - Part 3: Control actuators

EN 954-1: 1996 (ISO 13849-1:1999)

Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design

EN 60204-1:1997

Safety of machinery - Electrical equipment of machines - Part 1: General requirements

#### General standards

UL 508, 2001 (=CSA C22.2 No.14)

Industrial Control Equipment

EN 50178:1997

Electronic equipment for use in power installations

EN 61131-1:1994 (IEC1131-1)	Programmable controllers - Part 1: General information
EN 61131-2:1994 (IEC1131-2)	Programmable controllers - Part 2: Equipment requirements and tests
EN ISO 9241-10:1996	Ergonomic requirements for office work with visual display terminals (VDTs) - Part 10: Dialogue principles
<b>Use of robots with the KeTop</b>	
UL 1740, 1998	Industrial Robots and Robotic Equipment
ANSI/RIA R15.06, 1999 (=CSA-Z434-94)	ANS for Industrial Robots and Robot Systems - Safety Requirements
ANSI/RIA R15.02/1, 1990	ANS for Industrial Robots and Robot Systems - Hand-Held Robot Control Pendants - Human Engineering Design Criteria
ISO 10218:1992	Manipulating industrial robots - Safety
<b>Use of the KeTop in machining centers:</b>	
ISO 11161:1994	Industrial automation systems - Safety of integrated manufacturing systems - Basic requirements
EN 12417:2001	Machine tools - Safety - Machining centres
EN 14070:2003	Safety of machine tools - Transfer and special-purpose machines